

Oracle® Transportation Management

Administration Guide

Release 6.3

Part No. E38415-08

January 2015

Copyright © 2001, 2015, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Contents

CONTENTS.....	III
SEND US YOUR COMMENTS	IX
PREFACE	X
CHANGE HISTORY	X
1. ARCHITECTURE OVERVIEW	1-1
SERVER OVERVIEW	1-1
WEB SERVER	1-1
APPLICATION SERVER	1-2
DATABASE SERVER	1-2
INTEGRATION SERVER	1-2
2. CONFIGURING OPTIONAL COMPONENTS.....	2-1
CONFIGURING ORACLE TRANSPORTATION MANAGEMENT TO USE A REPLICATED ONLINE DATABASE.....	2-1
CONFIGURING ORACLE TRANSPORTATION MANAGEMENT TO USE ORACLE MAPVIEWER	2-1
CONFIGURING ORACLE TRANSPORTATION MANAGEMENT TO USE ORACLE SPATIAL.....	2-2
CONFIGURING CUSTOM EXTERNAL RATING AND DISTANCE ENGINES	2-2
CONFIGING ORACLE TRANSPORTATION MANAGEMENT TO SEND FAXES	2-2
CONFIGURING FAXMAKER FOR FAX NOTIFICATIONS	2-2
CONFIGURING RIGHTFAX FOR FAX NOTIFICATIONS	2-3
FAX GENERATION IN ORACLE TRANSPORTATION MANAGEMENT	2-4
CONFIGURING PCMILER WORLDWIDE OR PCMILER RAIL	2-5
CONFIGURING RAND McNALLY INTELLIRoute	2-6
CONFIGURING RAND McNALLY MILEMAKER	2-6
CONFIGURING SMC RATEWARE.....	2-7
CONFIGURING SMC RATEWAREXL SERVICE.....	2-7
CONFIGURING KEWILL FLAGSHIP.....	2-8
3. STARTING AND STOPPING THE ORACLE TRANSPORTATION MANAGEMENT SERVERS	3-1
STARTING ORACLE TRANSPORTATION MANAGEMENT SERVERS	3-1
STOPPING ORACLE TRANSPORTATION MANAGEMENT SERVERS.....	3-2
4. ADVANCED CONFIGURATION: CUSTOM PROPERTIES	4-1
PROPERTY TYPES.....	4-1
SINGLE-VALUED PROPERTIES	4-1
MULTI-VALUED PROPERTIES	4-1
PROPERTY MACROS	4-2
PROPERTY FILE MANAGEMENT	4-2
PROPERTY SETS	4-3
ENABLING PROPERTY SETS AFTER UPGRADE FROM RELEASES PRIOR TO 6.3.4	4-4

DISABLING PROPERTY SETS	4-5
MORE INFORMATION	4-5
5. ADVANCED CONFIGURATION: APPLICATION SERVER	5-1
CHANGING THE WEBLOGIC CONFIG.XML FILE	5-1
MANUALLY MAKE THE CHANGES TO THE "CONFIG.XML.FRESH" FILE	5-1
MAKE CHANGES USING THE WEBLOGIC CONSOLE	5-1
6. ADVANCED CONFIGURATION: UI/WEB SERVER	6-1
CREATING AND INSTALLING SECURE SOCKET LAYER (SSL) CERTIFICATES	6-1
ENABLING REVERSE-PROXY SUPPORT	6-1
ENABLING SINGLE SIGN ON (SSO) SUPPORT	6-1
ENABLING IP ADDRESS IN LOG-IN HISTORY.....	6-1
INSTALLING MULTIPLE WEB SERVERS	6-2
LOAD BALANCER.....	6-2
DIAGNOSTIC SCREENS.....	6-2
PREVENTING DOUBLE-LOGIN ISSUE.....	6-3
INSTALLING TRANSLATIONS.....	6-4
UNDER UNIX SYSTEMS:	6-4
UNDER MICROSOFT WINDOWS:	6-4
FUSION TRANSPORTATION INTELLIGENCE TRANSLATIONS	6-5
NOTIFICATIONS TRANSLATIONS	6-5
MODIFYING THE SESSION TIMEOUT	6-5
7. ADVANCED CONFIGURATION: DATABASE.....	7-1
WARNING AGAINST MODIFYING DATA IN DATABASE.....	7-1
CONFIGURING ORACLE TRANSPORTATION MANAGEMENT TO USE A DIFFERENT DATABASE	7-1
REAL APPLICATION CLUSTERS (RAC)	7-1
INTERNATIONAL CHARACTERS IN SQL*PLUS WINDOWS	7-2
SORTING	7-2
COPY DATABASE DATA.....	7-2
COPY DATABASE FILES	7-2
ORACLE DATABASE EXPORT/IMPORT UTILITY	7-3
DOMAIN COPY	7-4
USING WEBLOGIC GRIDLINK DATA SOURCE.....	7-4
CREATE DATA SOURCE IN WEBLOGIC CONSOLE.....	7-4
UPDATE DATABASE FOR DATA SOURCE	7-4
MODIFY STARTUP SCRIPTS ON SERVER	7-5
VALIDATE ORACLE TRANSPORTATION MANAGEMENT WITH EXTERNALLY DEFINED DATA SOURCE	7-6
8. CONFIGURING FUSION TRANSPORTATION INTELLIGENCE	8-1
CONFIGURING ORACLE TRANSPORTATION MANAGEMENT	8-1
SCHEDULING ETL THROUGH ORACLE TRANSPORTATION MANAGEMENT PROCESS MANAGEMENT.....	8-1
LOADING AN ORACLE TRANSPORTATION MANAGEMENT OBJECT INTO FUSION TRANSPORTATION INTELLIGENCE	8-1
UNLOADING AN ORACLE TRANSPORTATION MANAGEMENT OBJECT FROM FUSION TRANSPORTATION INTELLIGENCE ...	8-2
MANDATORY ORACLE TRANSPORTATION MANAGEMENT USER ROLE (VPD PROFILE) CONFIGURATION	8-2
POPULATING THE AD_TIME TABLE.....	8-3

CONFIGURING FUSION TRANSPORTATION INTELLIGENCE WITH OAM (SSO)	8-3
CONFIGURING ORACLE DATA INTEGRATOR (ODI)	8-3
ETL PERFORMANCE TUNING	8-3
CONFIGURING OBI EE	8-4
SETTING UP AN OBI EE AGENT	8-4
9. CONFIGURING GLOBAL TRADE INTELLIGENCE	9-1
CONFIGURING ORACLE TRANSPORTATION MANAGEMENT	9-1
SCHEDULING ETL THROUGH ORACLE TRANSPORTATION MANAGEMENT PROCESS MANAGEMENT	9-1
LOADING AN ORACLE GLOBAL TRADE MANAGEMENT OBJECT INTO GLOBAL TRADE INTELLIGENCE	9-1
UNLOADING A GLOBAL TRADE MANAGEMENT OBJECT FROM GLOBAL TRADE INTELLIGENCE	9-1
MANDATORY ORACLE TRANSPORTATION MANAGEMENT USER ROLE (VPD PROFILE) CONFIGURATION	9-1
POPULATING THE AD_TIME TABLE	9-2
CONFIGURING GLOBAL TRADE INTELLIGENCE WITH OAM (SSO)	9-2
CONFIGURING ORACLE DATA INTEGRATOR (ODI)	9-2
ETL PERFORMANCE TUNING	9-2
10. ADVANCED CONFIGURATION: GENERAL	10-1
GENERAL INSTRUCTIONS ON UPGRADING ORACLE TRANSPORTATION MANAGEMENT FROM ONE RELEASE TO ANOTHER	10-1
BRANDING	10-2
CUSTOMIZATIONS	10-2
CHANGING LOGGING OPTIONS	10-4
CHANGING NOTIFICATION SETTINGS	10-5
MULTI-BYTE CHARACTER SUPPORT IN BI PUBLISHER	10-5
UNITS OF MEASURE	10-5
CHANGING CURRENCY SETTINGS	10-6
WHEN TO CHANGE CURRENCY STORAGE DEFAULT	10-6
ACCOUNTING FOR MISSING TIME ZONES	10-7
COUNTRY CODE CONVERSION FROM ISO ALPHA-3 CODE TO ALPHA-2 CODE	10-8
DATA VALIDATION	10-8
PROCESS FLOW	10-9
APIS FOR COMPILING PKG_USER_EXIT	10-9
HTTP INTEGRATION	10-10
BPEL INTEGRATION	10-10
BPEL FLOWS	10-10
ORACLE TRANSPORTATION MANAGEMENT PROPERTIES FILES FOR ORACLE E-BUSINESS SUITE INTEGRATION THROUGH BPEL	10-10
ORACLE TRANSPORTATION MANAGEMENT PROPERTIES FILES FOR BPEL INTEGRATION	10-11
RECURRING PROCESSES	10-11
SMTP INTEGRATION	10-11
11. ADVANCED CONFIGURATION: LDAP	11-1
12. PERFORMANCE MONITORING	12-1
OPERATING SYSTEM MONITORING TOOLS	12-1

MICROSOFT WINDOWS	12-1
UNIX.....	12-2
ORACLE TRANSPORTATION MANAGEMENT TOOLS	12-4
CONFIGURATION DATA COLLECTION	12-4
PERFORMANCE DATA COLLECTION.....	12-4
WEB SERVER TOOLS.....	12-4
LOAD BALANCERS	12-4
APPLICATION SERVER TOOLS	12-4
ADMINISTRATION CONSOLE	12-4
JVM TOOLS.....	12-5
CONSOLE LOGS.....	12-5
THREAD DUMPS	12-5
HEAP DUMPS.....	12-5
CAPTURING JROCKIT MISSION CONTROL JRA FILES.....	12-5
DATABASE MONITORING TOOLS.....	12-6
AUTOMATIC WORKLOAD REPOSITORY.....	12-6
AUTOMATIC DATABASE DIAGNOSTIC MONITOR	12-6
NETWORK MONITORING.....	12-6
DIAGNOSING A PERFORMANCE ISSUE	12-6
STEPS FOR DIAGNOSING A PERFORMANCE ISSUE.....	12-6
ADDITIONAL STEPS FOR DATABASE RELATED ISSUE.....	12-7
13. PERFORMANCE TUNING.....	13-1
JVM	13-1
CHANGING MEMORY SETTINGS	13-1
WEB SERVER	13-2
SIMULTANEOUS USERS	13-2
APPLICATION SERVER.....	13-3
LOGGING	13-3
ORACLE TRANSPORTATION MANAGEMENT BUSINESS OBJECT CACHES	13-3
BUSINESS NUMBER GENERATOR (BNG) CONTENTION	13-4
DIRECTSQLUPDATE	13-5
WORKFLOW THREAD TUNING	13-5
AUTOMATIC TIMEOUT FOR STUCK THREADS.....	13-6
MULTI-THREADING BULK PLAN PROCESSING.....	13-6
DATABASE.....	13-6
ANALYZING TABLES/GATHERING STATISTICS	13-6
DATABASE FILE STORAGE PERFORMANCE.....	13-7
FULL EXPORT AND RE-IMPORT	13-7
CHECKING FOR STALE AND PARTIAL ORACLE DATABASE STATISTICS	13-8
MAINTAINING ORACLE DATABASE STATISTICS.....	13-8
SQL TUNING	13-8
DETERMINING THE ORIGIN OF A SQL STATEMENT	13-9
USING TIMEOUTS WITH SQL STATEMENTS.....	13-9
INIT.ORA PARAMETERS	13-10
TUNING A RAC DATABASE INSTALLATION USING SERVICES	13-10

HASH PARTITIONING INDEXES ON RAC DATABASES	13-11
CHANGING DATABASE POOL SIZE	13-12
PURGING EXCESS REPORTS FROM THE REPORT_LOG TABLE	13-12
PURGE PROCESS FOR WINDOWS	13-12
MANUALLY PURGING TRANSMISSIONS FROM THE DATABASE	13-13
PARTITIONED TABLES	13-14
PINNING LARGE OBJECT IN MEMORY	13-19
DATABASE SPACE MONITORING	13-20
SCHEDULED JOBS	13-20
OPTIMIZING APPLICATION-DATABASE ROUND TRIPS	13-20
INTEGRATION	13-21
SUPPRESSING PERSISTENCE XML	13-21
SYNCHRONIZATION OF MASTER DATA.....	13-21
OUTXMLPROFILE FOR OUTBOUND INTEGRATION AND NOTIFICATION	13-21
INBOUND INTEGRATION OPTIONS	13-22
USER INTERFACE	13-22
FINDER PAGE SIZE	13-22
USER FAVORITES	13-22
ADDING SEARCH FIELDS TO FINDERS (GRID-FLATTENING)	13-22
ACTIVE TABLE	13-23
STATUS QUERIES.....	13-23
MANAGER LAYOUT PRODUCER CONFIGURATION	13-24
PARAMETERIZED SAVED QUERIES	13-24
LIMITING CHILD RECORDS IN A MANAGER	13-24
FUNCTIONAL SECURITY CHECKACCESS CACHE SIZES.....	13-25
CLIENT SESSION MANAGEMENT	13-25
SCALABILITY	13-25
MISCELLANEOUS.....	13-25
OPTIMIZING REPORT GENERATION	13-25
14. TROUBLESHOOTING KNOWN ISSUES	14-1
INSTALLATION	14-1
503 (SERVICE UNAVAILABLE) ERRORS	14-1
UNINSTALLING ORACLE TRANSPORTATION MANAGEMENT	14-1
BROWSERS.....	14-1

Send Us Your Comments

Oracle Transportation Management Administration Guide, Release 6.3

Part No. E38415-08

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: otm-doc_us@oracle.com

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

If you have problems with the software, contact Support at <https://support.oracle.com> or find the Support phone number for your region at <http://www.oracle.com/support/contact.html>.

Preface

This document contains information regarding the post-install configuration and maintenance of the following products: Oracle Transportation Management, Global Trade Management, Fusion Transportation Intelligence and Oracle In-Memory Logistics Command Center. There are additional, Logistics Command Center-specific details in the Logistics Command Center Administration Guide.

Change History

Date	Document Revision	Summary of Changes
12/2012	-01	Initial release.
03/2013	-02	<p>Added note in the section "Truncate Jobs for Partitioned Tables" to notify users about I_TRANSACTION table's OUTBOUND5 partition, this partition does not get truncated by partit job.</p> <p>Added reference in Configuration FTI section to the Configuring FTI with OAM (SSO) that is now included in the OAM Integration Guide.</p> <p>Added a note to page 7-2. Note: The directories in the following section are controlled by the glog.customPath property. The directory defined by that property is represented by "custom" in the paths listed in the following seven subsections.</p>
06/2013	-03	New preface.
08/2013	-04	<p>Corrected properties in the Business Number Contention section in the Performance Tuning section.</p> <p>Updated eLocation descriptions.</p> <p>The start/stop section was changed to indicate that certain scripts are provided and may or may not be used.</p> <p>Removed Enterprise Java Beans section from the Application Server section</p> <p>The instances of "/servlets" have been replaced with "/GC3"</p>

Date	Docu ment Revision	Summary of Changes
12/2013	-05	<p>Reference to Enabling IP Address in Log-in History.</p> <p>Corrected reference to Migration Guide.</p> <p>Removed "Enabling Automatic Web UI Login" section from chapter 4.</p> <p>Added support for diagnostics across all web servers in a non-Scalable environment.</p> <p>New section added to the Admin Guide on Load Balancers.</p> <p>New section added on Configuring Global Trade Intelligence.</p> <p>Updated Configuring Fusion Transportation Intelligence section to include addition of GTI as a new product. This includes new VPD profile and properties.</p>
05/2014	-06	<p>Added new sub-section Unloading a Global Trade Management Object from Global Trade Intelligence</p> <p>Inserted the Data Validation section into Chapter 9.</p> <p>Added and revised sections dealing with property changes to document Property Set usage.</p> <p>Added section: Enabling Property Sets after Upgrade From Releases Prior to 6.3.4</p> <p>Added section: Disabling Property Sets</p>
09/2014	-07	<p>Introduced new database user DIR_XML_USER into the Oracle Transportation Management database. This user should be use for Direct XML integration. This user has minimum privileges to successfully insert XML transmissions into database objects when using Direct XML integration feature.</p> <p>Updated Units of Measure section to reflect the new UI.</p> <p>Updated Automatic Timeout for Stuck Threads section with corrected examples.</p>

Date	Document Revision	Summary of Changes
1/2015	-08	<p>Added section on HTTP Integration.</p> <p>Changed the SMTP Authentication section to be the SMTP Integration section.</p> <p>Updated Business Number Generator sequences to be: glog.server.bngenerator.oracleSequence.xid.SHIP_UNIT_XID.DEFAULT=SHIP_UNIT_GID_SEQUENCE glog.server.bngenerator.oracleSequence.xid.ORDER_RELEASE_LINE_XID.DEFAULT=ORDER_RELEASE_LINE_GID_SEQ</p> <p>Added section: "Using Timeouts with SQL Statements".</p> <p>Replaced supported databases with reference to Technical Architecture Guide.</p> <p>Removed language about BPEL being the preferred integration method.</p>

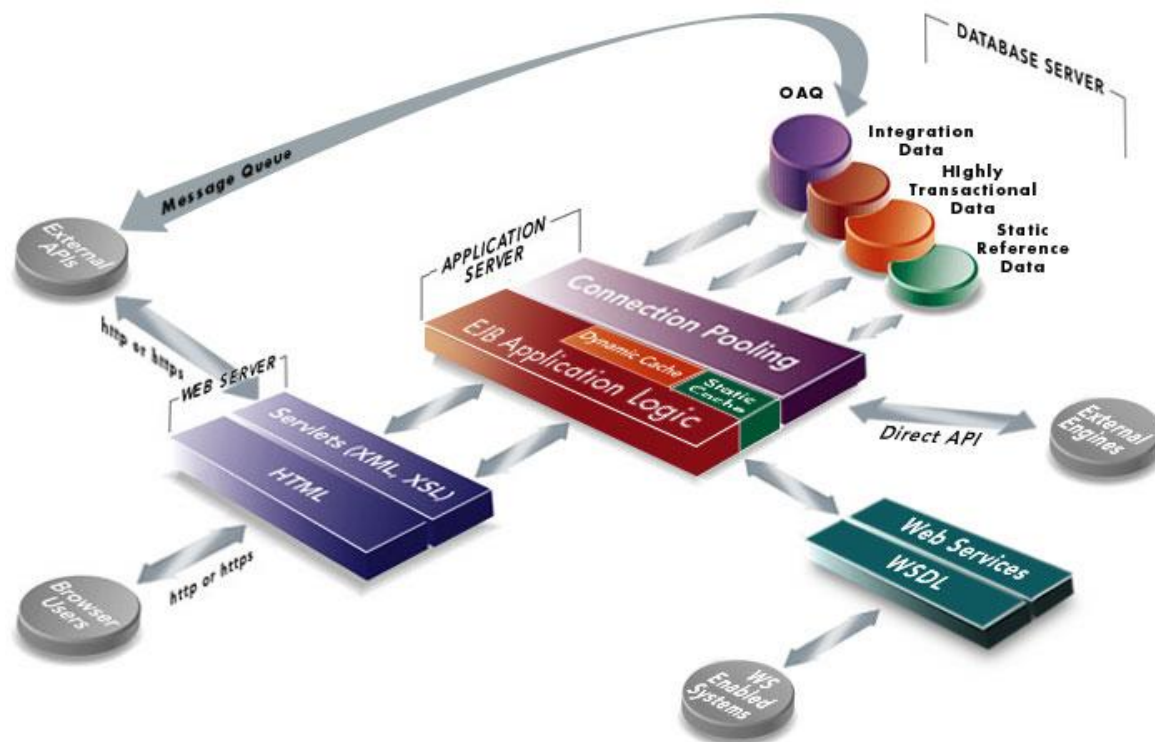
1. Architecture Overview

Oracle Transportation Management is built for interactive environments that leverage the internet as an information backbone to capture reference data such as updates to carriers, carrier rates, shipping schedules, locations, ports, and other logistics sources. The underlying architecture is a unique multi-enterprise data model with n-tier web and application servers, partitioned database server, native XML language, extensible style sheets, advanced data security, and optimal caching.

Oracle Transportation Management is highly distributed and collaborative. It is written in Java and uses Enterprise Java Beans, XML, HTML, and XSL technologies. The design is based on a thin HTML client/fat application architecture so that applications are easily deployable and accessible by multiple enterprises. It supports comprehensive integration via XML APIs to all of the application logic.

Server Overview

The following illustration describes the overall architecture of the Oracle Transportation Management environment.



Each server is designed to perform specific functions that are briefly described in the following sections. From a user's perspective, all that you need to use to access Oracle Transportation Management is a standard Web browser such as Microsoft Internet Explorer or Mozilla Firefox.

Web Server

The web server controls the Oracle Transportation Management user interface and communicates with the application server to initiate the appropriate application logic. The Oracle Transportation Management user interface is delivered to the user in HTML format and viewed using a browser such as Microsoft Internet Explorer or Mozilla Firefox.

The Oracle Transportation Management web server uses the Oracle HTTP Server (OHS) web server and the Tomcat Java servlet engine to manage the user interface and communications with the application server.

Application Server

The application server controls the application logic and communicates that logic between the web server and the Oracle Database. The application server runs Oracle WebLogic Server software to manage system-level details and operates at the center of the Oracle Transportation Management multi-tier architecture. In this architecture, business logic is executed in the application server, rather than in client applications. The resulting "thin" client, allows the client to manage the presentation layer, the application server to manage the business logic, and the back-end data services to manage the data.

Database Server

The database server is the central repository for all the data entered in Oracle Transportation Management. The database server communicates with the application server to deliver the appropriate data to the user based on the Oracle Transportation Management application logic. The database server also communicates with the integration server to handle the import and export of Oracle Transportation Management data with other systems.

Please see the Technical Architecture Guide for the supported databases.

Integration Server

The integration server is responsible for:

- Accepting Oracle Transportation Management XML from an Enterprise Application Integration (EAI) solution.
- Persisting XML transmission to a staging table.
- Accepting requests from the Oracle Transportation Management application server to build Oracle Transportation Management XML from staged data.
- Sending Oracle Transportation Management XML to an EAI solution.

The EAI solution is used as a secure data pipeline and data transformation engine and is responsible for:

- Sending and receiving EDI flat files to and from an EDI Van.
- Transforming EDI flat files to Oracle Transportation Management XML.
- Login security (communication is permitted only from registered IP addresses).
- Encryption security (using https).
- Guaranteed delivery (only if a partner server is on the remote side).

The integration server's job is to safely store XML transmissions and transform the XML transmissions into internal tables. The Oracle Transportation Management application server is responsible for initiating jobs that move data to and from staging tables and notifies the application server when new data has arrived in the staging table. The application server must notify the Integration server when it has moved new data to the staging table.

The integration server does not perform data validation. Data validation occurs when data is transferred from the staging table to the application server tables. The staging table does not enforce referential integrity or constraints.

The Integration server and the application server communicate to the application server using Enterprise Java Beans (EJB) session bean calls. When an integration event occurs, the integration server notifies the application server using a session bean call. When the application service needs to communicate with the outside world, it does so directly, using an integration solution that provides a method to build transmissions and forward them to the EAI solution.

2. Configuring Optional Components

These components include:

- Replicated Online Database (ROD)
- Oracle MapViewer server
- Oracle Spatial server
- Custom External Rating and Distance Engines
- GFI FAXmaker for Networks/SMTP 8.1
- RightFax
- PCMiller WorldWide & PCMiller Rail
- Rand McNally IntelliRoute Server
- SMC RateWare Server
- SMC RateWareXL Service
- Kewill FlagShip Server

Configuring Oracle Transportation Management to Use a Replicated Online Database

You can set up Oracle Transportation Management to use a Replicated Online Database (ROD) during the initial Oracle Transportation Management installation (as described in the Oracle Transportation Management Install Guide), or you can modify an existing installation to use a ROD. To modify an existing Oracle Transportation Management installation, you need to edit the `glog.properties` file on all Oracle Transportation Management application and web servers. This file is located in the `<otm_install_path>/glog/config` directory. Un-comment (if necessary) and modify the following lines, setting the values on the right as appropriate (see the relevant section of the Oracle Transportation Management Install Guide for details on what each field's value should be):

```
# Replicated Operational Database (ROD) - optional
ods_feature=true
glog.ods.dbserver=otmrod.company.com
glog.database.ods.sid=otmrod
glog.ods.connectstring=otmrod
glog.ods.database.port=1521
glog.database.ods.schema=glogowner
```

Once this change has been made, restart your Oracle Transportation Management instance.

Note that these changes for Oracle MapViewer may be made in the `APP_CUSTOM` Property Set rather than the `glog.properties` file directly. Once completed, you must still restart your Oracle Transportation Management instance. See Section 4 for more information.

Configuring Oracle Transportation Management to Use Oracle MapViewer

You can set up Oracle Transportation Management to integrate with Oracle MapViewer server during the initial Oracle Transportation Management installation (as described in the Oracle Transportation Management Install Guide), or you can modify an existing installation to use an Oracle MapViewer server. To modify an existing Oracle Transportation Management installation, you need to edit the `glog.properties` file on all Oracle Transportation Management application servers. This file is located in the `<otm_install_path>/glog/config` directory. Un-comment (if necessary) and modify the

following lines, setting the values on the right as appropriate (see the relevant section of the Oracle Transportation Management Install Guide for details on what each field's value should be):

```
# Oracle MapViewer server
glog.mapServer=mapviewer.company.com
glog.map.service_name=otm_world
glog.map.basemap=otm_world.world_sample
```

Once this change has been made, restart your Oracle Transportation Management instance.

Note that these changes for Oracle Spatial may be made in the `APP_CUSTOM` Property Set rather than the `glog.properties` file directly. Once completed, you must still restart your Oracle Transportation Management instance. See Section 4 for more information.

Configuring Oracle Transportation Management to Use Oracle Spatial

You can set up Oracle Transportation Management to integrate with an Oracle Spatial server during the initial Oracle Transportation Management installation (as described in the Oracle Transportation Management Install Guide), or you can modify an existing installation to use an Oracle Spatial server. To modify an existing Oracle Transportation Management installation, you need to edit the `glog.properties` file on all Oracle Transportation Management application servers. This file is located in the `<otm_install_path>/glog/config` directory. Un-comment (if necessary) and modify the following lines, setting the values on the right as appropriate (see the relevant section of the Oracle Transportation Management Install Guide for details on what each field's value should be):

```
# Oracle Spatial server
OracleSpatial.host=otmspatial.company.com
OracleSpatial.port=7777
OracleSpatial.US_Canada.routeServlet=/routeserver/servlet/RouteServerSe
rvlet
OracleSpatial.WesternEurope.routeServlet=/routeserver_eu/servlet/RouteS
erverServlet
OracleSpatial.geocodeServlet=/geocoder/gcserver
```

Once this change has been made, restart your Oracle Transportation Management instance.

Configuring Custom External Rating and Distance Engines

This section has been broken out into its own document. Please see the **External Programming Interface Guide** for more information.

Configuring Oracle Transportation Management to Send Faxes

Configuring FAXmaker for Fax Notifications

FAXmaker is an email-to-fax gateway that allows Oracle Transportation Management to send out fax notifications. Oracle Transportation Management sends an email via SMTP to a specific POP3 mailbox. FAXmaker checks this mailbox at regular intervals and converts the emails within to fax transmissions, if all security requirements are satisfied.

1. Install FAXmaker for Networks/SMTP 8.1 from GFI on a Microsoft Windows server and configure it to work with your modem(s). Refer to the FAXmaker documentation for details.
1. Test the modem to ensure that it gets a dial tone and can access an outside line.

2. Create a POP3 mailbox within your mail server that can be accessed by the FAXmaker server. The mailbox name must be identical to the email address for outgoing fax notifications that you entered during the Oracle Transportation Management application server install. Test this POP3 account using any email client that supports POP3. Please contact your System Administrators for more information as they will have greater insight into the makeup of your SMTP mail services.
3. Install the FAXmaker software according to the vendor's instructions.
4. Copy the Oracle Transportation Management fax template file (`<otm_install_path>\install\FAXmaker\G-Log.rtf` or `<otm_install_path>/install/FAXmaker/G-Log.rtf`) to FAX maker's coverage directory (e.g. `C:\Program Files\FAXmaker\Coverpage\`).
5. Please note the following:
 - a. Ensure that your modem is configured to send faxes ONLY; do not configure the modem to allow faxes to be received.
 - b. Ensure that G-Log.rtf is the default coverpage and set the paper size to letter.
 - c. Ensure that any special dialing requirements (such as dialing 7 to get an outside line) are set according to the vendor's documentation.
 - d. The username and email address of the user you add to FAXmaker must be identical to the email address from which email notifications will come that you entered when you installed the Oracle Transportation Management application server. This allows emails that originate from this email address to be converted and faxed out. If an email is received from an email address that does not have an account, FAXmaker will bounce it.
6. Refer to the FAXmaker documentation for any troubleshooting assistance or contact GFI directly.

You can set up Oracle Transportation Management to integrate with FAXmaker during the initial Oracle Transportation Management installation (as described earlier), or you can modify an existing installation to use FAXmaker. To modify an existing Oracle Transportation Management installation, you need to edit the `glog.properties` file on all Oracle Transportation Management application servers. This file is located in the `<otm_install_path>/glog/config` directory. Un-comment (if necessary) and modify the following lines, setting the values on the right as appropriate (see the relevant section of the Oracle Transportation Management Install Guide for details on what each field's value should be):

```
# FaxMaker Settings - optional
glog.workflow.notify.faxmaker.email=fax@company.com
```

Once this change has been made, restart your Oracle Transportation Management instance.

Note that this property change for FAXmaker may be made in the `APP_CUSTOM` Property Set rather than the `glog.properties` file directly. Once completed, you must still restart your Oracle Transportation Management instance. See Section 4 for more information.

Configuring RightFax for Fax Notifications

RightFax is an email-to-fax gateway that allows Oracle Transportation Management to send out fax notifications. The Oracle Transportation Management application sends an email via SMTP to a specific POP3 mailbox. RightFax checks this mailbox at regular intervals and converts the emails within to fax transmissions, if all security requirements are satisfied.

1. Install RightFax 8.5 from Captaris on a Microsoft Windows 2000 server and configure it to work with your modem(s). Refer to the RightFax documentation for details.

2. Test the modem to ensure that it gets a dial tone and can access an outside line.
3. Create a POP3 mailbox within your mail server that can be accessed by the RightFax server. The mailbox name must be identical to the email address for outgoing fax notifications that you entered during the Oracle Transportation Management application server install. Test this POP3 account using any email client that supports POP3. Please contact your System Administrators for more information as they will have greater insight into the makeup of your SMTP mail services.
4. Install RightFax according to the vendor's installation guide and configure it to poll the POP3 mailbox created above.

You can set up Oracle Transportation Management to integrate with RightFax during the initial Oracle Transportation Management installation (as described earlier), or you can modify an existing installation to use RightFax. To modify an existing Oracle Transportation Management installation, you need to edit the `glog.properties` file on all Oracle Transportation Management application servers. This file is located in the `<otm_install_path>/glog/config` directory. Uncomment (if necessary) and modify the following lines, setting the values on the right as appropriate (see the relevant section of the Oracle Transportation Management Install Guide for details on what each field's value should be):

```
# RightFax Settings - optional
glog.fax.defaultHandler.routingMode=to
glog.fax.defaultHandler.routingPicture=/name={lastName},{firstName}/fax
={rawPhone}/
glog.fax.email=fax@company.com
```

Once this change has been made, restart your Oracle Transportation Management instance.

Note that these changes for RightFax may be made in the `APP_CUSTOM` Property Set rather than the `glog.properties` file directly. Once completed, you must still restart your Oracle Transportation Management instance. See Section 4 for more information.

Fax Generation in Oracle Transportation Management

Fax generation is controlled via a plug-and-play interface, `FaxHandler`. Classes that implement this interface support the following two methods:

```
public MailAddress getAddress();
// returns the mail address for fax generation

public void handle(MailEvent event, FaxTopic faxTopic);
// modifies our standard text e-mail to conform to outgoing fax
// specifications.
```

The fax handler is specified via the property `glog.fax.handler=<fax handler class name>`

Oracle Transportation Management supplies a standard fax handler, `glog.server.workflow.notify.DefaultFaxHandler`, to support most fax generation software. This handler embeds routing information into one of the following 3 mail components:

- message body
- subject line
- to alias (i.e. the last name of the to party)

The embedding is accomplished with a picture string describing the modified format of the above component. You can specify original information from the text email to use in the modified component including:

- firstName
- lastName
- company
- phone
- rawPhone (punctuation or space)
- message
- subject

Replacements are represented by one of these identifiers surrounded by parenthesis; for example, {firstName}. Newline characters are preserved. To transform an email message body to one supporting FAXMaker, the picture string would be:

```
::{firstName},{company},{lastName},,{rawPhone}\n{message}
```

The following properties control the default fax handler:

```
glog.fax.defaultHandler.routingMode=<message|body|to> (default:
message)
glog.fax.defaultHandler.routingPicture=<picture string>
(default:
::{firstName},{company},{lastName},,{rawPhone}\n{message})
```

For RightFax, the properties default to:

```
glog.fax.defaultHandler.routingMode=to
glog.fax.defaultHandler.routingPicture=/name={lastName},
{firstName}/fax={rawPhone}/
glog.fax.email=fax@company.com
```

For FAXMaker, they default to:

```
glog.fax.defaultHandler.routingMode=message
glog.fax.defaultHandler.routingPicture>::{firstName},{company},{lastNam
e},,{rawPhone}\n{message}
glog.fax.email=fax@company.com
```

Note that these changes for default fax handling may be made in the APP_CUSTOM Property Set rather than the glog.properties file directly. See Section 4 for more information.

Configuring PCMiler WorldWide or PCMiler Rail

Install the PCMiler product as described in your PCMiler user's manual. Install the PCMiler Canada Postal Codes, if desired (PCMiler WorldWide only).

You can set up Oracle Transportation Management to integrate with PCMiler during the initial Oracle Transportation Management installation (as described in the Oracle Transportation Management Install Guide), or you can modify an existing installation to use PCMiler.

Note: PCMiler must be configured to return its distance in miles.

To modify an existing Oracle Transportation Management installation, you need to edit the glog.properties file on all Oracle Transportation Management application servers. This file is located in the <otm_install_path>/glog/config directory. Uncomment (if necessary) and modify the following lines, setting the values on the right as appropriate (see the relevant section of the Oracle Transportation Management Install Guide for details on what each field's value should be):

```
pcmiler.host=pcmiler.company.com  
pcmiler.port=8145
```

Uncomment and modify the following lines for PCMiller Rail:

```
pcmiler.rail.host=pcmilerrail.company.com  
pcmiler.rail.port=2001
```

Once this change has been made, restart your Oracle Transportation Management instance.

Note that these changes for PCMiller products may be made in the `APP_CUSTOM` Property Set rather than the `glog.properties` file directly. Once completed, you must still restart your Oracle Transportation Management instance. See Section 4 for more information.

Configuring Rand McNally IntelliRoute

Note: Due to lack of support for 64-bit Java Development Kit (JDK), IntelliRoute is not currently supported as of the time of this writing. Please contact Rand McNally for an ETA on when it will be available.

Install the Rand McNally IntelliRoute server as described in your IntelliRoute user's manual. You will also need to create a user (e.g. `otm`), password, and location for the Oracle Transportation Management application to use. These activities are described in the IntelliRoute user's manual. Once the server is installed, you need to install the IntelliRoute Java API onto the Oracle Transportation Management application server as described in your IntelliRoute user's manual.

You can set up Oracle Transportation Management to integrate with IntelliRoute during the initial Oracle Transportation Management installation (as described in the Oracle Transportation Management Install Guide), or you can modify an existing installation to use IntelliRoute. To modify an existing Oracle Transportation Management installation, you need to edit the `glog.properties` file on all Oracle Transportation Management application servers. This file is located in the `<otm_install_path>/glog/config` directory. Uncomment (if necessary) and modify the following lines, setting the values on the right as appropriate (see the relevant section of the Oracle Transportation Management Install Guide for details on what each field's value should be):

```
intelliroute.host=otmir.company.com  
intelliroute.port=1998  
intelliroute.user=otm  
intelliroute.password=changeme  
intelliroute.location=company1
```

Once this change has been made, restart your Oracle Transportation Management instance.

Note that these changes for Intelliroute may be made in the `APP_CUSTOM` Property Set rather than the `glog.properties` file directly. Once completed, you must still restart your Oracle Transportation Management instance. See section 4 for more information.

Configuring Rand McNally MileMaker

Install the MileMaker server as described in your user's manual.

You can set up Oracle Transportation Management to integrate with MileMaker during the initial Oracle Transportation Management installation (as described in the Oracle Transportation Management Install Guide), or you can modify an existing installation to use MileMaker. To modify an existing Oracle Transportation Management installation, you need to edit the `glog.properties` file on all Oracle Transportation Management application servers. This file is located in the `<otm_install_path>/glog/config` directory. Uncomment (if necessary) and modify the following

lines, setting the values on the right as appropriate (see the relevant section of the Oracle Transportation Management Install Guide for details on what each field's value should be):

```
milemaker.host=otmmm.company.com  
milemaker.port=1031
```

Once this change has been made, restart your Oracle Transportation Management instance.

Note that these changes for MileMaker may be made in the `APP_CUSTOM` Property Set rather than the `glog.properties` file directly. Once completed, you must still restart your Oracle Transportation Management instance. See Section 4 for more information.

Configuring SMC RateWare

Install the SMC RateWare server and load the CZAR tariffs as described in your RateWare user's manual.

You can set up Oracle Transportation Management to integrate with RateWare during the initial Oracle Transportation Management installation (as described in the Oracle Transportation Management Install Guide), or you can modify an existing installation to use RateWare. To modify an existing Oracle Transportation Management installation, you need to edit the `glog.properties` file on all Oracle Transportation Management application servers. This file is located in the `<otm_install_path>/glog/config` directory. Uncomment (if necessary) and modify the following lines, setting the values on the right as appropriate (see the relevant section of the Oracle Transportation Management Install Guide for details on what each field's value should be):

```
glog.RatingEngine.Rateware.URL=otmsmc.company.com  
glog.RatingEngine.Rateware.port=23700
```

Once this change has been made, restart your Oracle Transportation Management instance.

Note that these changes for RateWare may be made in the `APP_CUSTOM` Property Set rather than the `glog.properties` file directly. Once completed, you must still restart your Oracle Transportation Management instance. See Section 4 for more information.

Configuring SMC RateWareXL Service

You can set up Oracle Transportation Management to integrate with the RateWareXL Service during the initial Oracle Transportation Management installation (as described in the Oracle Transportation Management Install Guide), or you can modify an existing installation to use the RateWareXL Service. To modify an existing Oracle Transportation Management installation, you need to edit the `glog.properties` file on all Oracle Transportation Management application servers. This file is located in the `<otm_install_path>/glog/config` directory. Uncomment (if necessary) and modify the following lines, setting the values on the right as appropriate (see the relevant section of the Oracle Transportation Management Install Guide for details on what each field's value should be):

```
# external SMC RatewareXL - optional  
glog.RatingEngine.RatewareXL.Username=SomeUserName  
glog.RatingEngine.RatewareXL.Password=SomePassword  
glog.RatingEngine.RatewareXL.License=qwerty12345
```

Once this change has been made, restart your Oracle Transportation Management instance.

Note that these changes for RateWareXL may be made in the `APP_CUSTOM` Property Set rather than the `glog.properties` file directly. Once completed, you must still restart your Oracle Transportation Management instance. See Section 4 for more information

Configuring Kewill FlagShip

Install the Kewill FlagShip server as described in your FlagShip user's manual.

You can set up Oracle Transportation Management to integrate with Kewill FlagShip during the initial Oracle Transportation Management installation (as described in the Oracle Transportation Management Install Guide), or you can modify an existing installation to use Kewill FlagShip. To modify an existing Oracle Transportation Management installation, you need to edit the `glog.properties` file on all Oracle Transportation Management application servers. This file is located in the `<otm_install_path>/glog/config` directory. Uncomment (if necessary) and modify the following lines, setting the values on the right as appropriate (see the relevant section of the Oracle Transportation Management Install Guide for details on what each field's value should be):

```
glog.RatingEngine.Kewill.URL=otmkewill.company.com
glog.RatingEngine.Kewill.port=1200
glog.RatingEngine.Kewill.RefnumQual=KEWILL_CUST_NUM
```

Once this change has been made, restart your Oracle Transportation Management instance.

Note that these changes for FlagShip may be made in the `APP_CUSTOM` Property Set rather than the `glog.properties` file directly. Once completed, you must still restart your Oracle Transportation Management instance. See Section 4 for more information.

3. Starting and Stopping the Oracle Transportation Management Servers

Starting Oracle Transportation Management Servers

Oracle provides shell scripts for use with UNIX operating systems to start and stop the various components. These scripts may be used as-is or may be modified as needed (in which case it is recommended that you save the modified files to a different filename). The shell scripts are suitable for use with most `/etc/init.d`-type automatic startup environments. After installation these shell scripts can be found at:

- `<otm_install_dir>/install/ohs/glogweb-wl`
- `<otm_install_dir>/install/weblogic/glogapp-wl`

Note: These scripts will be referred to in the instructions below by their names as given above. Please adjust the instructions given below as needed to reflect any name changes or location changes of the scripts being used for your specific environment.

To start Oracle Transportation Management, please do the following:

1. Start the Oracle Transportation Management database, including any listeners
2. Start the Oracle Transportation Management web server
 - a. On Microsoft Windows:
 - i. Start the "Server Manager".
 - ii. Expand "Configuration" and then click on "Services".
 - iii. Locate the Oracle HTTP Server (OHS) service (e.g. "Oracle Process Manager (instance1)" – see the OHS documentation for details), right-click on it and select "Start" ; wait 10 seconds before proceeding to the next step.
 - iv. Locate the Oracle Transportation Management Tomcat service (e.g. `otmwebtomcat63`), right-click on it and select "Start"; wait 10 seconds before proceeding to the next step.
 - b. On UNIX systems:
 - i. Start the Oracle Transportation Management web daemon (e.g. `<otm_install_dir>/install/ohs/glogweb-wl start`)
3. Start the Oracle Transportation Management application server
 - a. On Microsoft Windows:
 - i. Start the "Server Manager"
 - ii. Expand "Configuration" and then click on "Services"
 - iii. Locate the Oracle Transportation Management WebLogic service (e.g. `otmapp63`), right-click on it and select "Start"
 - b. On UNIX systems:
 - i. Start the Oracle Transportation Management app server daemon (e.g. `<otm_install_dir>/install/weblogic/glogapp-wl start`)
 - c. It can take several minutes for the application server to fully start up, depending on your hardware
 - d. You can monitor the `<otm_install_path>/logs/weblogic/console.log.0` file to determine when WebLogic has been fully initialized. Look for lines similar to the following:

```

INFO | 2012/11/12 10:38:16 | <Nov 12, 2012 10:38:16 AM PST> <Notice>
<Server> <BEA-002613> <Channel "Default" is now listening on
10.241.18.128:7001 for protocols iiop, t3, ldap, snmp, http.>
INFO | 2012/11/12 10:38:16 | <Nov 12, 2012 10:38:16 AM PST> <Notice>
<Server> <BEA-002613> <Channel "DefaultSecure" is now listening on
10.241.18.128:7002 for protocols iiops, t3s, ldaps, https.>
INFO | 2012/11/12 10:38:16 | <Nov 12, 2012 10:38:16 AM PST> <Notice>
<WebLogicServer> <BEA-000329> <Started WebLogic Admin Server "gc3-
slc02jwd" for domain "Otmv630" running in Production Mode>
INFO | 2012/11/12 10:38:17 | <Nov 12, 2012 10:38:17 AM PST> <Notice>
<WebLogicServer> <BEA-000365> <Server state changed to RUNNING>
INFO | 2012/11/12 10:38:17 | <Nov 12, 2012 10:38:17 AM PST> <Notice>
<WebLogicServer> <BEA-000360> <Server started in RUNNING mode>

```

4. INFO | 2012/11/12 10:38:26 | -- OTM Event: serverReady To check to see if everything has started up, point to `http://<web server name>` in your web browser and log in.

Note: While the application server is starting up, the web server will respond to all requests with a '503 (Service Unavailable)' error message. This will go away once the application server has fully started up.

Stopping Oracle Transportation Management Servers

To start Oracle Transportation Management, please do the following:

5. Stop the web server
 - a. On Microsoft Windows:
 - i. Start the "Server Manager".
 - ii. Expand "Configuration" and then click on "Services".
 - iii. Locate the Oracle Transportation Management Tomcat service (e.g. `otmwebtomcat63`), right-click on it and select "Stop".
 - iv. Locate the Oracle HTTP Server (OHS) service (e.g. "Oracle Process Manager (instance1)" – see the OHS documentation for details), right-click on it and select "Stop".
 - b. On UNIX systems:
 - i. Stop the Oracle Transportation Management web daemon (e.g. `<otm_install_dir>/install/ohs/glogweb-wl stop`)
6. Stop the application server
 - a. On Microsoft Windows:
 - i. Start the "Server Manager"
 - ii. Expand "Configuration" and then click on "Services"
 - iii. Locate the Oracle Transportation Management WebLogic service (e.g. `otmapp63`), right-click on it and select "Stop"
 - b. On UNIX systems:
 - i. Stop the Oracle Transportation Management application daemon (e.g. `<otm_install_dir>/install/weblogic/glogapp-wl stop`)
7. Stop the Oracle database, if necessary.

If you are going to restart Oracle Transportation Management, wait one minute between stopping and restarting the servers. This gives the application server the time it needs to fully shut down.

4. Advanced Configuration: Custom Properties

Much of the configuration and customization of Oracle Transportation Management involves managing properties used by the application. These properties are initially distributed in a set of hierarchical property files, where one property file can include another. This allows reuse of common properties on both web servers and application servers and the overriding of these staged properties for a particular installation.

Property Types

To understand property precedence, it's important to consider two basic types of properties in Oracle Transportation Management:

- Single-valued. This is a property that has a single value for a given key.
- Multi-valued. This is a property that supports multiple values for a given key.

While most configurable properties are single-valued, a list of multi-valued properties can be found in `glog.readonly.properties` under `glog.properties.multiple` entries. The `glog.workflow.topicGroup` property, for example, is multi-valued: each value declares the number of threads to allocate to a workflow topic queue.

Single-Valued Properties

Assume a single-valued property is defined in more than one property file and on more than one line in a given property file. The declaration in the top-most property file takes precedence over declarations in included files. The declaration lower in a property file takes precedence over one defined above it. As an example, assume we have the following property files:

glog.properties

```
a.property=value1
!include standard.properties
a.property=value3
```

standard.properties

```
a.property=value2
```

The value of **a.property** is set to **value3** since the values in `glog.properties` take precedent over those in `standard.properties` and the line with **value3** is below the line with **value1**. Only if both declarations in `glog.properties` were removed would the value of **a.property** be set to **value2**.

Multi-Valued Properties

Multi-valued properties work in a more additive way. Assume that **a.property** was declared to be multi-valued. Then the values in both `glog.properties` and `standard.properties` are used. The values for **a.property** will be **value1**, **value2** and **value3**. If you needed to override (i.e. remove) **value2**, use the **!remove** directive. This directive can be used to remove all values declared to date or to remove a specific value. In the above example, we can remove **value2** as follows:

glog.properties

```
a.property=value1
!include standard.properties
!remove a.property=value2
a.property=value3
```

standard.properties

```
a.property=value2
```

The resulting list of values for **a.property** is now **value1** and **value3**.

Workflow Example

A common example of multi-valued property use is the specification of thread count for a workflow queue. The multi-valued properties that define workflow threads are given by:

```
glog.workflow.topicGroup=<queue name>,<thread count>
```

By monitoring workflow queue backup, you may need to increase the default thread count for a specific queue. The **batch** queue, for example, handles many long-running batch processes. Assume you wanted to increase the default thread count from 2 to 6. Simply adding the property:

```
glog.workflow.topicGroup=batch,6
```

is insufficient because it leaves the initial property declaration in place. The recommended approach is to remove the initial declaration and then add the new one:

```
!remove glog.workflow.topicGroup=batch,2
glog.workflow.topicGroup=batch,6
```

Note that the **!remove** directive only works on properties that have been loaded. It must be declared after the initial value has been included.

Property Macros

A property macro is a property whose value is used in the values for other properties. Any property can be used by another via the following syntax:

```
<property>=[<prefix>]$<macro>$[<suffix>]
```

where **<macro>** is a previously declared property. When evaluating property macros, the property macro must have been declared prior to its use. E.g., assume you have the following property files:

glog.properties

```
a.property=$a.macro$
!include standard.properties
another.property=$a.macro$
```

standard.properties

```
a.macro=test
```

The evaluation of **a.property** has no value for **a.macro**. It is set to '**\$a.macro\$**'. But the value for **another.property** is set properly to **test**.

Property File Management

Oracle Transportation Management ships with a set of property files in the `<otm_install_path>/glog/config` directory, comprising the property heirarchy. The top-level file of this hierarchy is `glog.properties`.

When instructed to change a property value, you will be instructed to make changes to only the web server, only the application server, or both. In any case, be sure to make your changes to the `glog.properties` file. All property changes made in any file other than `glog.properties` will be over-written during every upgrade.

When adding, removing or modifying properties in the webserver's `glog.properties` file, make your changes after the section starting with:

```
#-----
# Custom Properties - Beginning
#-----
# Place all changes, new properties, and custom properties here.
# This section will be used during GC3 Upgrades
#-----
```

and before:

```
#-----
# Custom Properties - End
#-----
```

This ensures property macros and multi-valued properties that you may need have been declared before your custom changes are processed.

Similarly, when adding, removing or modifying properties in the application server's `glog.properties` file, make all non-workflow changes after the section starting with:

```
#-----
# Custom Properties - Beginning
#-----
# Place all changes, new properties, and custom properties here.
# This section will be used during GC3 Upgrades
#-----
```

and before:

```
#-----
# Custom Properties - End
#-----
```

Changes to workflow thread counts must be made in a section below the Custom Properties section to ensure they override previously declared thread counts. Make these changes after the section starting with:

```
#-----
# Custom Thread Properties - Beginning
#-----
# Place all thread changes and properties here.
# This section will be used during GC3 Upgrades
#-----
# This directive allows for the use of Property Set
APP_WORKFLOW_THREADING to manage
# workflow groups and their thread counts. This property set is stored
in the
# OTM database and is accessible through the Property Set manager
within OTM.
# Comment this out to disable this data-driven property set.
```

and before:

```
#-----
# Custom Thread Properties - End
#-----
```

Property Sets

Most properties used by Oracle Transportation Management can be customized using *Property Sets*, instead of property files. A property set is a collection of property instructions residing in the database

rather than in a file. Property sets can be modified via the Property Set Manager. By default, this is available only to the **DBA.ADMIN** user via **Configuration and Administration > Property Management > Property Sets**.

Property sets support these types of instructions:

- **Set.** Set the value of a property
- **Remove.** Remove all or one value of a multi-valued property
- **Include.** Include another property set

Note that an included property set can be marked as a *child*, where properties in the parent set take precedence over properties in the included set; or a *sibling*, where properties in the included set take precedent over properties in the parent¹

Four property sets are included in the standard Oracle Transportation Management installation:

- **APP_CUSTOM.** This holds custom application server properties.
- **WEB_CUSTOM.** This holds custom web server properties.
- **CUSTOM.** This holds custom properties shared by the web server and application server.
- **APP_WORKFLOW_THREADING.** This holds custom thread counts for workflow queues.

Each of these property sets can be used to enter custom properties via the Property Set Manager rather than directly in the file. This can be useful in an environment where business administrators have no access to the properties files but need to make property changes that impact various logic within the application.

From the standpoint of property sets, there are three types of property lifecycles:

- **Reserved.** These properties impact the ability for the system to properly restart and read the database. They cannot be modified in property sets.
- **Restart.** These properties can be modified in property sets, but won't take effect until the servers are restarted. A list of properties pending restart, by server, can be accessed via **Configuration and Administration > Property Management > Pending Properties**.
- **Dynamic.** These properties can be modified in property sets and take immediate effect.

The list of reserved and restart properties can be viewed in the `glog.readonly.properties` file.

Enabling Property Sets after Upgrade from Releases Prior to 6.3.4

When migrating from versions prior to 6.3.4, property sets are not enabled by default. Entries in the **APP_CUSTOM**, **WEB_CUSTOM**, **CUSTOM** and **APP_WORKFLOW_THREADING** property sets are ignored.

To enable property sets for implementations migrated from these older releases:

1. Edit the `glog.properties` file for each web server and insert the line
`!propertySet WEB_CUSTOM,sibling`
before the line

¹ assuming the parent properties are declared before the property set inclusion

```
!include glog.webserver.properties
```

2. Edit the `glog.properties` file for each application server and insert the line

```
!propertySet WEB_CUSTOM,sibling
```

before the line:

```
!include glog.webserver.properties
```

Also insert the line:

```
!propertySet APP_CUSTOM,sibling
```

before the line:

```
!include glog.appserver.properties
```

Finally, insert the line:

```
!propertySet APP_WORKFLOW_THREADING,sibling
```

before the lines:

```
#-----  
# Custom Thread Properties - End  
#-----
```

Disabling Property Sets

Property sets can be disabled without modifying property files through the use of system properties. This can be useful when running a command-line tool that cannot access the database due to a firewall setup, but needs other OTM properties. To disable all property set loading, set the following system variable when invoking OTM or one of its tools:

```
-Dglog.propertySets.enabled=false
```

It may also be useful to allow property sets but suppress any errors generated by missing property sets. If a property set is being designed, for example, it may be helpful to add it first to the property file via a `!propertySet` directive and stage it later in the database. To suppress the default errors logged to the console, set

```
-Dglog.propertySets.reportErrors=false
```

More Information

For more information and complete documentation of all properties, please consult the online help topic "Property Files".

5. Advanced Configuration: Application Server

Note: Default paths are used below and may differ from your configuration.

Changing the WebLogic `config.xml` File

WebLogic uses a file called "`config.xml`" to store and read configuration data. This file is generated automatically every time the application is started, using a file called "`config.xml.fresh`" as the basis. Changes made directly to the "`config.xml`" file, either manually by the user or through the WebLogic console, will therefore be overwritten and lost whenever the application is started. In order to preserve the changes, do one of the following whenever changes need to be made:

Manually Make the Changes to the "`config.xml.fresh`" File

1. Back up the original "`config.xml.fresh`" file first:

```
$ cd <otm_install_dir>/weblogic/domains/otm
$ cp config.xml.fresh config.xml.fresh_20130417_1421
```
2. Make the changes directly to the "`config.xml.fresh`" file using the editor of your choice.
3. Restart the application.

Make Changes Using the WebLogic Console

1. Back up the original "`config.xml.fresh`" file first:

```
$ cd <otm_install_dir>/weblogic/domains/otm
$ cp config.xml.fresh config.xml.fresh_20130417_1421
```
2. Make the necessary changes via the WebLogic console and click on "Activate".
3. Copy the new "`config.xml`" file to "`config.xml.fresh`":

```
$ cd <otm_install_dir>/weblogic/domains/otm
$ cp config.xml config.xml.fresh
```
4. Edit the "`config.xml.fresh`" file in the editor of your choice and remove the "<jdbc-system-resource>" section for the PRIMARY_JTS connection; it will look something like this:

```
<jdbc-system-resource>
  <name>PRIMARY_JTS</name>
  <target>gc3-slc01ebb</target>
  <descriptor-file-name>jdbc/PRIMARY_JTS-8636-
jdbc.xml</descriptor-file-name>
</jdbc-system-resource>
```
5. Delete those lines and save the file.

6. Advanced Configuration: UI/Web Server

Note: Default paths are used below and may differ from your configuration.

Note: Whenever you edit the Oracle Transportation Management property files, only edit the `glog.properties` file. Any changes that you make to any other files will be overwritten every time an update is applied.

Creating and Installing Secure Socket Layer (SSL) Certificates

This information has moved to the Oracle Transportation Management Security Guide.

Enabling Reverse-Proxy Support

Oracle Transportation Management supports some reverse-proxy solutions. The reverse-proxy is expected to identify sites based on a URL-prefix (such as `/otm`) and then strip that prefix off before forwarding the URL to Oracle Transportation Management. Every link that Oracle Transportation Management sends to the browsers will contain the defined URL-prefix.

To enable reverse-proxy support, do the following²:

1. Edit the `<otm_install_path>/glog/config/glog.properties` file on your web and application servers and modify the line to include the desired URL-prefix (ex. `/otm`):
`glog.webserver.urlprefix=`
2. Edit the `index.htm` file under `<otm_install_path>/web/htdocs`. The following line needs to contain the URL-prefix:
`<META http-equiv="refresh" content="0; URL=/GC3/glog.webserver.servlet.umd.Login"/>`
 - For example, if your URL-prefix is `/otm`, it would look like:
`<META http-equiv="refresh" content="0; URL=/otm/GC3/glog.webserver.servlet.umd.Login"/>`
3. Ensure that the `glog.webserver.URL` setting in your `glog.properties` files points to your reverse-proxy server, rather than the individual web server(s).
4. Restart Oracle Transportation Management to enable these changes. Oracle Transportation Management can now be accessed through the reverse-proxy, but may not be accessible directly. This is a limitation due to adjusting links to work with the reverse-proxy.

Enabling Single Sign On (SSO) Support

This information has moved to the Oracle Transportation Security Guide.

Enabling IP Address in Log-in History.

This information exists in the Oracle Transportation Management Security Guide.

² The properties listed here are reserved and cannot be set in a property set.

Installing Multiple Web Servers

Multiple web servers can be utilized to increase the performance of Oracle Transportation Management. Generally, with more web servers you can maintain more simultaneous user connections into the Oracle Transportation Management server's web interface. You may also see increased performance in integration, since incoming integration files are posted to a servlet on the web server and are passed back to the application server.

Load Balancer

We recommend using a hardware load balancer to spread the incoming requests among multiple web servers as efficiently as possible. The load balancer should be configured so that once a user logs into a given web server, that same web server is always given back to that user within some configurable time-period. After that time has elapsed, the web server given back to the user can once again be determined by current load, after which the server returned should once again be fixed for a given time-period. This is commonly referred to as a persistent or sticky session, and is usually accomplished by inserting a cookie or other header into the request. It can also be done by IP address, though this is a less desirable alternative. Exact configurations and options depend on the load balancer in use.

Some load balancers can also be configured to dynamically remove a web server from the rotation if it is down for whatever reason, and dynamically add it back in when it recovers. This is usually done by having the load balancer attempt to connect to a specific URL on the web server every N seconds or minutes, and there is a servlet provided to handle this specifically. Configure your load balancer to load the `glog.webserver.test.TestServlet` servlet, e.g.:

```
http://www.company.com/GC3/glog.webserver.test.TestServlet
```

If the servlet loads properly and returns a response code of 200, then everything is fine. If any other response code is given, or if a connection cannot be established, then that should be considered an error condition and the server should be dropped out of rotation. Test out the URL manually with a browser on a known-working system before using it to configure the load balancer; it should return the word "OK" on the screen.

Diagnostic Screens

Oracle Transportation Management supports a number of diagnostic pages and processes that display configuration information and monitor web-tier system health, including:

- Log browsing
- Properties management
- Cache monitoring
- JMS monitoring
- Scalability balancing

If the installation is customized to support application-tier scalability³, diagnostic screens are aware of all web servers. That is, from a single browser request, the screens retrieve data across all web servers.

In an installation with multiple web servers but a single application server – one without application-tier scalability activated – the diagnostic screens can still be configured to retrieve and report on data across all web servers. To accomplish this, the installation must inform the application server and each

³ See the Application Scalability Guide for more information on this configuration.

web server of all potential web servers. To do this you will need to add properties to inform each web server of the existence of other web servers. To do this directly in the "glog.properties" file:

1. Back up the existing "glog.properties" file.

```
$ cd <otm_install_dir>/glog/config  
$ cp glog.properties glog.properties_20130417_2156
```
2. Edit the "glog.properties" file.
3. Locate the section that looks like this:

```
!remove glog.scalability.web.topologyMachineURL  
glog.scalability.web.topologyMachineURL=<HTTP URL for webserver #1>
```

and add a property for each Web server:

```
!remove glog.scalability.web.topologyMachineURL  
glog.scalability.web.topologyMachineURL=<HTTP URL for webserver #1>  
glog.scalability.web.topologyMachineURL=<HTTP URL for webserver #2>  
...  
glog.scalability.web.topologyMachineURL=<HTTP URL for webserver #n>
```

4. Restart OTM.
Note: In the case where a Web server has more than one URL (e.g. an internal URL and an external URL), the URL listed should be one that is accessible from the machine whose "glog.properties" file is being edited.

Alternatively, this can be done by modifying the `WEB_CUSTOM` property set, setting the `glog.scalability.web.topologyMachineURL` property for each web server.

Preventing Double-Login Issue

If a load-balancer is used, the Tomcat configuration file will need to be adjusted to prevent an issue where a user has to login twice. This can also happen in some reverse proxy configurations. The problem presents itself as:

- The user logs in via the load balancer URL.
- After successful login, the user is redirected to another login page, this time with the URL of a specific web server (not the load balancer).
- After the second successful login, the user may use the system, but only by hitting the web server directly, not through the load balancer.

The solution is to use the load balancer's information in the Tomcat "server.xml" configuration file. To do that, follow these steps:

1. Back up the existing "server.xml" file.

```
$ cd <otm_install_dir>/tomcat/conf  
$ cp server.xml server.xml_20130417_2156
```
2. Edit the "server.xml" file.
3. Locate the <Connector> element and change the proxyName and proxyPort attributes:

Before

```
<Connector address="localhost" port="8009" protocol="HTTP/1.1"  
            connectionTimeout="3600000"  
            proxyName="otmweb.company.com" proxyPort="80"  
            URIEncoding="UTF-8" />
```

After

```
<Connector address="localhost" port="8009" protocol="HTTP/1.1"
connectionTimeout="3600000"
proxyName="otm-lb.company.com" proxyPort="80"
URIEncoding="UTF-8" />
```

4. Restart the web server instance.

Installing Translations

Under UNIX systems:

1. Log in as the Oracle Transportation Management user.
2. Change the directory to <otm_install_path>/glog/oracle/script8.
3. Run the following command: `install_lang.sh <lang id> [<country id>]`.

Under Microsoft Windows:

1. Log in as the Oracle Transportation Management user.
2. Start > Run... > `cmd`.
3. Change the directory to <otm_install_path>/glog/oracle/script8.
4. Run the following command: `install_lang.sh <lang id> [<country id>]`

where valid <lang id> and <country id> values are:

Language	Language ID	Country ID
Chinese (Simplified)	zh	CN
Chinese (Traditional)	zh	TW
French (Canadian)	fr	CA
French (European)	fr	
German	de	
Italian	it	
Japanese	ja	
Korean	ko	
Portuguese (Brazilian)	pt	BR
Russian	ru	
Spanish (European)	es	
Spanish (Latin American)	es	MX
Turkish	tr	

Fusion Transportation Intelligence Translations

Oracle Fusion Transportation Intelligence translations are installed when the Fusion Transportation Intelligence application is installed; please see the Installing Other Languages for Oracle Fusion Transportation Intelligence Reports section of the Oracle Transportation Management Install Guide for more information.

Notifications Translations

To have notifications translated, set the following property in the application server's `glog.properties` file or the `APP_CUSTOM` property set.:

```
glog.notify.localizeStylesheet=true
```

Modifying the Session Timeout

This information has been moved to the Oracle Transportation Management Security Guide.

7. Advanced Configuration: Database

Note: Default paths are used below and may differ from your configuration.

Note: Whenever you edit the Oracle Transportation Management property files, only edit the `glog.properties` file. Any changes that you make to any other files will be overwritten every time an update is applied.

Warning Against Modifying Data in Database

Except as explicitly noted in this and other Oracle Transportation Management documentation, the data in an Oracle Transportation Management database should *never* be added to, modified or deleted by an outside application while Oracle Transportation Management is running. This could lead to unrecoverable data corruption and is not a supported use-case.

Configuring Oracle Transportation Management to Use a Different Database

At some point, you may need to point your Oracle Transportation Management instance to another database for testing, development, or fail-over purposes. Follow the directions below to configure Oracle Transportation Management to use this new database. It is assumed that the new Oracle Transportation Management database has already been created and populated with data using the database installation directions earlier in the Installation Guide. The database should also be at the same patch level as your Oracle Transportation Management instance to prevent any incompatibility problems.

1. Shutdown your Oracle Transportation Management web and application servers as described earlier in this guide.
2. Edit the file `<otm_install_path>/glog/config/glog.properties` and change the following properties. This should be done on each Oracle Transportation Management web and application server.

```
dbserver=<db_server_fqdn>
glog.database.sid=<oracle_sid>
glog.database.connectstring=<oracle_connectstring>
glog.database.port=<oracle_port>
```

Note that these properties are reserved and cannot be set as properties in a property set.

3. Restart your Oracle Transportation Management web and application servers as described earlier in this guide.
4. You may also need to edit your `tnsnames.ora` files on each Oracle Transportation Management server.

Real Application Clusters (RAC)

Oracle Transportation Management may be used in conjunction with RAC by using the SCAN Listener for the RAC instance in the connection information stored in the `<otm_install_path>/glog/config/glog.properties` file on all application and web servers:

```
dbserver=<SCAN listener host info>
glog.database.sid=<SCAN listener service name>
glog.database.connectstring=<SCAN listener TNS alias>
glog.database.port=<SCAN listener port>
```

Note that these properties are reserved and cannot be set as properties in a property set.

Special configuration in the RAC instance is needed to use a SCAN Listener. Please see the Oracle Real Application Cluster documentation for more details.

International Characters in SQL*Plus Windows

Oracle Transportation Management supports multiple languages. However, you can only display characters for one language at a time. The following needs to be done to properly configure this application to work in another language:

- Specify NLS_LANG as a system or environment variable on your client. The NLS_LANG parameter has three components: language, territory and charset. Examples of NLS_LANG settings are:

```
NLS_LANG = AMERICAN_AMERICA.WE8ISO8859P1  
NLS_LANG = JAPANESE_JAPAN.JA16EUC
```

For more information on NLS_LANG see the Oracle National Language Support Guide.

Important: The character set part of your NLS_LANG setting needs to correspond to your client application character set. For example, in an English Microsoft Windows environment, the usual character set of GUI applications is the operating system code page 1252, which corresponds to the Oracle character set WE8MSWIN1252. Therefore you would set your NLS_LANG to something like AMERICAN_AMERICA.WE8MSWIN1252. An incorrect NLS_LANG setting in this case would be AMERICAN_AMERICA.WE8ISO8859P15.

- If characters still do not display properly, then you should switch to a font that contains support for the characters you need. You can find more information on how to do this from Oracle. For example, http://otn.oracle.com/docs/tech/sql_plus/content.html.

Note: To find out which font supports the characters/script you need to use the Character Map utility on Microsoft Windows Server.

Sorting

Proper sorting in languages other than English can currently not be done via the Oracle Transportation Management web interface. However, language-sensitive sorting of Oracle Transportation Management data is possible when you have direct access to the Oracle Transportation Management database with applications such as Toad and SQL*Plus by setting the desired sort order as an ALTER SESSION parameter. For example, by issuing the command ALTER SESSION SET NLS_SORT = 'SWEDISH'; the sort order for the current session will be changed to Swedish.

Copy Database Data

When moving from initial implementation to production, or duplicate production to development, there are needs to copy the whole database or part of Oracle Transportation Management data between production and development/test environment. Depending on the purpose and requirement, copying an Oracle Transportation Management database/data can be achieved using different methods. It is recommended that Support be consulted to decide the best way of moving data.

Copy Database Files

An Oracle Database can be cloned by copying datafiles and other configuration files to the destination environment (usually another server). Procedures of this method can be found from Oracle Database support/document.

Features/Limitations:

- Simple and straightforward. There is no need to pre-create an Oracle Database in the destination machine.
- Fast. The total time needed is governed by physically transferring datafiles from source machine to destination machine.
- Whole database copy. If there are data in the source database that the client does not want to copy to the destination database, then data cleaning process should be carried out in the destination database after the copy.
- Source and destination servers must be the same platform, same operating system and same Oracle Database version between.
- Requires DBA skills.
- Requires down time in the source database if there is no hot-backup.

Oracle Database Export/Import Utility

Using the Oracle Database export/import method, the entire Oracle Transportation Management database structures can be copied from one database to another. To successfully copy an Oracle Transportation Management database, following guidelines are recommended:

1. Create all tablespaces used by Oracle Transportation Management first in the target database.
2. Check if all of Oracle Transportation Management required init.ora parameters are properly set in the target database.
3. Oracle Transportation Management database roles and schema users should be created in the target database before the import. This should be done by running Oracle Transportation Management supplied SQL scripts as described below.
4. Log in to the target database as user **SYS**.
`@create_glog_users.sql -- Make sure no errors`
5. Perform export on the source database. The easiest way is to do a FULL export. But if for any reason you do not want a FULL export, you can do a schema export. Oracle supplies a schema export parameter file (exp_GC3.par), where it lists all schemas the Oracle Transportation Management application needs.
6. Perform import on the target database. You can do a FULL import if the export was a FULL export. Alternatively you can do a schema import. Oracle supplies a schema import parameter file (imp_GC3.par), where it lists all schemas the Oracle Transportation Management application needs.
7. After the import finished successfully, there are post-import SQL scripts to be run in the target database. The steps to run these scripts are described in the Installation Guide in the Database Server section. It is important to follow the instructions to run the scripts. Ensure you log in to the database as the correct users when running the scripts.

Features/Limitations:

- Can be used between different platforms and different operating systems.
- Oracle Database version in the destination environment must be equal or higher than that of source environment.
- Need to pre-create an Oracle Database in the destination environment.
- Whole Oracle Transportation Management database copy. If there are data in the source database that the client does not want to be copied to the destination database, then data cleaning process should be carried out in the destination database after the import.
- May take a long time to finish depending on the size of the database.

Domain Copy

Oracle Transportation Management provides utilities to copy domains between Oracle Transportation Management databases. There are two approaches to copy domains. The first one uses a PL/SQL procedure to generate INSERT statements with domain data to be copied from the source database. The insert statements can be run in the target database. The second approach uses Oracle Database TABLE mode export/import with WHERE clause to move domain data between source and destination databases. Please refer to the Oracle Transportation Management document titled "Data Management Guide" for details of this method.

Features/Limitations:

- Can be used between different platforms and different operating systems.
- Oracle Database version in the target environment must be equal or higher than that in the source environment if using export/import approach.
- Requires pre-creating an Oracle Database and installing an Oracle Transportation Management database in the target environment.
- Oracle Transportation Management table structures must be identical between the source and target database.
- Can just move one or more domain data between the databases.

Using WebLogic GridLink Data Source

Oracle Transportation Management can be configured to use GridLink Data Source defined in WebLogic.

Note: This configuration requires you to stop and start Oracle Transportation Management.

In order to set up to use an external data source, you need to do the following steps. Oracle Transportation Management may not start if all steps are not completed.

1. Create Data Source in WebLogic Console
2. Update Database for Data Source
3. Modify startup scripts on server

Create Data Source in WebLogic Console

- Navigate to the WebLogic console.
- Login as system user.
- Follow WebLogic guidelines to Configure GridLink for RAC Data Source.
 - DO NOT create the Data Source name with prefix PRIMARY.
 - Remember the name of the Data Source and the JNDI name of this Data Source. In this example, the name of the Data Source is CUSTOM_DS and the JNDI name of the Data Source is CUSTOM

Note: You will see an existing Data Source named PRIMARY_JTS. You can ignore this connection.

Update Database for Data Source

Create a new data source and assign data functions to this data source using SQL scripts.

Note: See the Changing Passwords section of the Oracle Transportation Management Security Guide for more information on encoding password for user glogdba.

```

insert into data_source
(data_source_gid,data_source_xid,data_database_gid,manager,jdbc_driver,jdbc_url,schema,oracle_user,oracle_password,vpd_user,is_profiling_leaks,is_multi_user,domain_name) values
('CUSTOM_DS','CUSTOM_DS','GC3','EXTERNAL
JTS','oracle.jdbc.OracleDriver','jndi:CUSTOM','glogowner','glogdba',
'<encoded password>','DBA.ADMIN','N','Y','PUBLIC');

update data_source_profile_d set data_source_gid = 'PRIMARY_THIN'
where data_function_gid = 'STARTUP';

update data_source_profile_d set data_source_gid = 'CUSTOM_DS' where
data_source_gid = 'PRIMARY_JTS';

commit;

```

Modify startup scripts on server

Note: If you restart Oracle Transportation Management instance at this point, it will overwrite your changes made in WebLogic console. Then you have to repeat the steps for creating Data Source in WebLogic.

Single Oracle Transportation Management Application Server Instance

- Log into the server machine as the Oracle Transportation Management install user.
- Shut down the Oracle Transportation Management Application server.
- Go to the directory <OTM_INSTALL>/weblogic and modify config.xml.fresh as described below.
- Copy jdbc-system-resource element from config.xml to config.xml.fresh (Create a backup copy of config.xml prior to performing this step.)
 1. Copy config.xml.fresh config.xml.fresh.old.
 2. Open config.xml and search for the jdbc-system-resource element.
 3. Copy the entire jdbc-system-resource element and paste into config.xml.fresh.
- Restart the Oracle Transportation Management Application server.

jdbc-system-resource element:

```

<jdbc-system-resource><name>%Data Source
Name%</name><target>%host_appname%</target>
<descriptor-file-name>jdbc/%jdbc config%.xml</descriptor-file-name>
</jdbc-system-resource>

```

Example:

```

<jdbc-system-resource><name>CUSTOM</name><target>gc3-
slc01qra</target>
<descriptor-file-name>jdbc/CUSTOM-8305-jdbc.xml</descriptor-file-
name>
</jdbc-system-resource>

```

Note: If the descriptor-file-name has a prefix of PRIMARY, then the jdbc configuration will be deleted by the weblogic.sh file during the next startup.

Multiple Oracle Transportation Management Application Server Instances

The new data source created using WebLogic console will not be available for all Oracle Transportation Management application instances. Repeat the above steps for each application server with same data source and JNDI name, target value will be different according to application server.

Note: Do not copy `config.xml.fresh` or the jdbc configuration file from one server to another. The target name has to match the target name for the specific Oracle Transportation Management instance.

Validate Oracle Transportation Management with Externally Defined Data Source

If Oracle Transportation Management does not start, then follow instructions to validate and reset the Oracle Transportation Management Data Source configuration.

- The new data source should not have any records in the CONNECTION_POOL table.
 - This query should return 0 rows. Replace the custom data source name to match the name defined in OTM earlier

```
select count(1) from connection_pool where data_source_gid =
'CUSTOM_DS' ;
```
- WebLogic console has options to test the Data Source. This test should be successful.
- In WebLogic console, you should not see the PRIMARY_JTS Data Source after a restart.

Reset Configuration

If there are errors and Oracle Transportation Management does not start, use these instructions to reset the configuration.

- Shutdown the Oracle Transportation Management instance.
- Run the following SQL scripts:

```
update data_source_profile_d set data_source_gid = 'PRIMARY_JTS'
where data_source_gid = 'CUSTOM_DS';

delete from data_source where data_source_gid = ' CUSTOM_DS';

commit;
```
- Restore the backup of `config.xml.fresh`. This file should not have any JDBC configuration
- Start the Oracle Transportation Management instance.

8. Configuring Fusion Transportation Intelligence

Configuring Oracle Transportation Management

Scheduling ETL through Oracle Transportation Management Process Management

You can schedule the ETL to run using the Run ETL process in Oracle Transportation Management process manager. Run ETL run the ETL from the replicated online database (ROD) or the online transaction processing (OLTP) database to the historic database (HD).

Since OTM now supports several business intelligence applications, there are properties that must be set to specify which business intelligence application the ETL loads. Depending on the following properties, the run ETL will either load either the GTI or FTI tables or both:

- `allow_advanced_analytics` is the property that controls if FTI is running. If this is set to TRUE, ETL loads FTI tables only.
- `isAllowedGTIANalytics` is the property that control if GTI is running. If this is set to TRUE, ETL loads GTI tables only.
- if both properties are set to TRUE, then ETL loads both FTI and GTI tables.

Running the ETL

1. Log into Oracle Transportation Management as DBA.ADMIN.
2. Go to **Transportation Intelligence > Process Management > Run ETL**.
3. Schedule/run the Run ETL process.
4. Check the logs for errors.

Note: Before running the ETL, you must complete the steps mentioned in the following section "Enabling Oracle Fusion Transportation Intelligence Agents in Oracle Transportation Management" to enable loading objects into Fusion Transportation Intelligence.

Loading an Oracle Transportation Management Object into Fusion Transportation Intelligence

For Oracle Transportation Management business objects to be loaded into Fusion Transportation Intelligence, they must have a status type similar to `##_HD_LOAD_STATUS` with a status value of `##_READY_TO_LOAD`. For example, order base with a statue type of `OB_HD_LOAD_STATUS` and a status value of `OB_READY_TO_LOAD`. By default, this status type is set to a value of `##_NOT_READY_TO_LOAD`.

Note: `##_` is object code, Shipment code is SH, Order Base code is OB and Order Release code is OR.

Setting the Status for Existing Objects:

The Oracle Transportation Management installation provides a sample script to set the status on existing objects. This script is located at:

```
<otm_install_path>/glog/oracle/script8/advanced_analytics/load_hd_status.sql.
```

You can customize this script based on your business process and run it in OLTP database by logging in with glogowner credentials.

Setting the Status for New Objects:

Oracle Transportation Management ships with several PUBLIC automation agents to set the status of shipments, order bases, and order releases to `##_READY_TO_LOAD` to allow business objects to load into Fusion Transportation Intelligence. You need to enable these automation agents as described below.

Enabling Oracle Fusion Transportation Intelligence Agents in Oracle Transportation Management

To enable the automation agents required to load Oracle Transportation Management business objects into Fusion Transportation Intelligence, complete the following:

1. Log on to Oracle Transportation Management as DBA.ADMIN.
2. Go to **Business Process Automation > Agents and Milestones > Automation Agent**.
3. Search for and activate the following automation agents:
 - `LOAD_ORDER_BASE_TO_HD` (Default Event: Order base created)
 - `LOAD_ORDER_RELEASE_TO_HD` (Default Event: Order on shipment tendered)
 - `LOAD_SHIPMENT_TO_HD` (Default Event: Shipment tendered)
4. Change the event if necessary.

Unloading an Oracle Transportation Management Object from Fusion Transportation Intelligence

When an object is deleted from Oracle Transportation Management, you can also remove it from Fusion Transportation Intelligence. You can use the following agent actions to mark objects as soft deleted in the HD database.

- `UNLOAD SHIPMENT FROM HD`
- `UNLOAD ORDER RELEASE FROM HD`
- `UNLOAD ORDER BASE FROM HD`
- `UNLOAD ORDER MOVEMENT FROM HD`
- `UNLOAD SELL SHIPMENT FROM HD`
- `UNLOAD QUOTE FROM HD`
- `UNLOAD INVOICE FROM HD`
- `UNLOAD BULK PLAN FROM HD`
- `UNLOAD ORDER ITEM FROM HD`

You can create automation agents using the agent actions listed above. An example of an automation agent which marks order releases as soft deleted in the HD database is shown below:

1. Select an Agent Type of `ORDER RELEASE`.
2. Select an Agent Event of `ORDER - REMOVED` with Restrictions of `INTEGRATION`, `INTERNAL`, or `USER`. You do not need to select the Pre-persist option on the Restrictions page.
3. Add a single agent action of `UNLOAD ORDER RELEASE FROM HD`.

Mandatory Oracle Transportation Management User Role (VPD Profile) Configuration

Identify the Oracle Fusion Transportation Intelligence users. Assign the appropriate Business Intelligence Role to each user.

Oracle Fusion Transportation Intelligence offers additional external predicates from version 6.2. These are available under the following standard VPD Profiles shipped with the Oracle Transportation Management application:

- **FTI_DEFAULT:** This VPD profile is applicable for all Oracle Fusion Transportation Intelligence users who are not service providers in Oracle Transportation Management. This includes all the external predicates available in the existing DEFAULT Oracle Transportation Management VPD profile and the new external predicates specific to the Oracle Fusion Transportation Intelligence solution's Historical Database tables.
- **SERVPROV:** This VPD Profile is applicable for all Oracle Fusion Transportation Intelligence users who are also service providers in Oracle Transportation Management. This includes all the external predicates available in the existing SERVPROV Oracle Transportation Management VPD profile and the new external predicates specific to the Oracle Fusion Transportation Intelligence solution's Historical Database tables.

As a result, it is mandatory that you update the user role for all your existing Oracle Fusion Transportation Intelligence users to include either the FTI_DEFAULT or SERVPROV VPD Profiles, as applicable. Performing this manual step is mandatory to ensure the proper operational behavior of the Oracle Fusion Transportation Intelligence application.

If you are using a customized VPD profile in Oracle Transportation Management, alternatively you must manually include your additional external predicates over the FTI_DEFAULT or SERVPROV VPD profiles as applicable.

Populating the AD_TIME Table

The AD_TIME table is not populated by the Fusion Transportation Intelligence installation. If you want to use fiscal time as a dimension, you must populate the AD_TIME table as follows:

1. Populate the following columns in the AD_TIME table in the Oracle Transportation Management database schema. These columns correlate standard calendar dates to the fiscal dates.

a. FISCAL_YEAR	VARCHAR2(50)
b. FISCAL_QUARTER_ID	VARCHAR2(50)
c. FISCAL_MONTH_ID	VARCHAR2(50)
d. FISCAL_WEEK_ID	VARCHAR2(50)
e. FISCAL_DAY	VARCHAR2(50)
2. Run the ETL.

Configuring Fusion Transportation Intelligence with OAM (SSO)

Refer to the Configuring FTI with OAM (SSO) in the OAM Integration Guide for details on setting up single-sign on for Fusion Transportation Intelligence with OAM.

Configuring Oracle Data Integrator (ODI)

ETL Performance Tuning

When you have very large OLTP/ROD source databases, such as ETL source, there is a chance of the ETL running slowly. You can complete the steps listed below to improve ETL performance.

Note: These steps may slow down inserts/updates of data, so only complete this section when it is absolute necessary to optimize ETL.

Creating Indexes

First, you create indexes as follow:

1. Connect to the source database as glogowner and run the following:
`<OTM_INSTALL_HOME>/script8/fti_indexes.sql`
2. Connect to the source database using a sys user and run following commands:
`alter system set optimizer_index_caching=0 scope=BOTH;`
`alter system set optimizer_index_cost_adj=100 scope=BOTH;`
3. Install and configure the ODI Agent in the target (hdowner) database instance machine.

Changing the ODI Agent's Heap Size

Next, you change the heap size of the ODI agent as follows:

1. Open one of the following files depending on the operating system:
 - a. Windows: `<ODI_HOME>/oracledi/agent/bin/odiparams.bat`
 - b. Linux: `/oracledi/agent/bin/odiparams.sh`.
2. Change following parameters values based on available RAM.
`set ODI_INIT_HEAP=512m`
`set ODI_MAX_HEAP=1024m`
3. Go to **Topology Manager > Physical Architecture > Oracle > ORACLE_OTM/ORACLE_FTI**.
4. Increase Array Fetch Size/Batch Update Size based on the environment. You can increase the sizes to 100 and check the performance.
5. If you observe no improvement in performance, then increase to some other numbers and try again.
6. Continue to increase these numbers until you see performance improvement. Eventually the performance increase will level off.

Configuring OBI EE

Setting up an OBI EE Agent

To set up an OBI EE Delivers Agent, complete the following:

Note: In Fusion Transportation Intelligence, these are called Alerts.

1. In OBI EE enterprise manager, set up the SMTP server. Refer to the OBI EE documentation for details.
2. All Business Intelligence Roles that ship with Oracle Transportation Intelligence have the correct privileges to access Oracle Delivers and create agents (alerts). However, if an Oracle Transportation Management user cannot access the Alerts page, you need to enable alerts as follows:
 - a. Log into Oracle Transportation Management as DBA.ADMIN.
 - b. Click **Transportation Intelligence > Alerts**.
 - c. Click the **Administration** link in the upper right corner of the Oracle Business Intelligence page.
 - d. Under **Security**, click **Manage Privileges**.
 - e. Under **Access**, make sure the Business Intelligence Role specified on your Oracle Transportation Management user is listed next to **Access to Delivers**.

- f. Log out of Oracle Transportation Management.
- 3. Log back into Oracle Transportation Management as the user that will create alerts.
- 4. Click **Transportation Intelligence > Alerts**.
- 5. In the upper right of the Oracle Business Intelligence page, click the **down arrow next to Signed In As USER.ADMIN** and click **My Account**.
- 6. In the **My Account** page, click **Deliver Options** tab.
- 7. Set up a **device** and **delivery profile**. Refer to the Oracle Business Intelligence help for details.
- 8. Click **OK** to save the device.
- 9. Follow the tabs to **set up an agent**. Refer to the Oracle Business Intelligence help and/or documentation for details.

9. Configuring Global Trade Intelligence

Configuring Oracle Transportation Management

Scheduling ETL through Oracle Transportation Management Process Management

You can schedule the ETL to run using the Run ETL process in Oracle Transportation Management process manager. Run ETL run the ETL from the replicated online database (ROD) or the online transaction processing (OLTP) database to the historic database (HD).

Since OTM now supports several business intelligence applications, there are properties that must be set to specify which business intelligence application the ETL loads. Depending on the following properties, the run ETL will either load either the GTI or FTI tables or both:

- `allow_advanced_analytics` is the property that controls if FTI is running. If this is set to TRUE, ETL loads FTI tables only.
- `isAllowedGTIANalytics` is the property that control if GTI is running. If this is set to TRUE, ETL loads GTI tables only.
- if both properties are set to TRUE, then ETL loads both FTI and GTI tables.

Running the ETL

1. Log into Oracle Transportation Management as DBA.ADMIN.
2. Go to **Global Trade Intelligence > Process Management > Run ETL**.
3. Schedule/run the Run ETL process.
4. Check the logs for errors.

Note: All of the GTM objects included in GTI are automatically loaded.

Loading an Oracle Global Trade Management Object into Global Trade Intelligence

By default, the Global Trade Management objects supported by GTI are all loaded into GTI. Please see the Data Flow to Global Trade Intelligence online help topic for complete details.

Unloading a Global Trade Management Object from Global Trade Intelligence

You can unload (soft delete) data from the Global Trade Intelligence historical database (HD). When data is deleted from GTM/OTM, you can mark that record as deleted in the Global Trade Intelligence HD. The record remains in the HD, but is it filtered out using the GTI_DEFAULT VPD profile.

There are several PUBLIC automation agents and agent actions intended for use with Global Trade Intelligence to enable this functionality. Please see the Data Flow to Global Trade Intelligence online help topic for complete details.

Mandatory Oracle Transportation Management User Role (VPD Profile) Configuration

Identify the Oracle Global Trade Intelligence users. Assign the appropriate Business Intelligence Role to each user.

Oracle Global Trade Intelligence offers additional external predicates. These are available under the following standard VPD Profile shipped with the Oracle Transportation Management application:

- **GTI_DEFAULT:** This VPD profile is applicable for all Oracle Global Trade Intelligence users. This includes all the external predicates available in the existing DEFAULT and FTI_DEFAULT Oracle Transportation Management VPD profiles and the new external predicates specific to the Oracle Global Trade Intelligence solution's Historical Database tables.

As a result, it is mandatory that you update the user role for all your existing Oracle Fusion Transportation Intelligence users to include the GTI_DEFAULT profiles, as applicable. Performing this manual step is mandatory to ensure the proper operational behavior of the Oracle Global Trade Intelligence application.

If you are using a customized VPD profile in Oracle Transportation Management, alternatively you must manually include your additional external predicates over the GTI_DEFAULT VPD profile as applicable.

Populating the AD_TIME Table

The AD_TIME table is not populated by the Global Trade Intelligence installation. If you want to use fiscal time as a dimension, you must populate the AD_TIME table as follows:

1. Populate the following columns in the AD_TIME table in the Oracle Transportation Management database schema. These columns correlate standard calendar dates to the fiscal dates.

a. FISCAL_YEAR	VARCHAR2(50)
b. FISCAL_QUARTER_ID	VARCHAR2(50)
c. FISCAL_MONTH_ID	VARCHAR2(50)
d. FISCAL_WEEK_ID	VARCHAR2(50)
e. FISCAL_DAY	VARCHAR2(50)
2. Run the ETL.

Configuring Global Trade Intelligence with OAM (SSO)

Refer to the Configuring FTI/GTI with OAM (SSO) in the OAM Integration Guide for details on setting up single-sign on for Global Trade Intelligence with OAM.

Configuring Oracle Data Integrator (ODI)

ETL Performance Tuning

When you have very large OLTP/ROD source databases, such as ETL source, there is a chance of the ETL running slowly. You can complete the steps listed in ETL Performance Tuning to improve ETL performance.

Note: These steps may slow down inserts/updates of data, so only complete this section when it is absolute necessary to optimize ETL.

10. Advanced Configuration: General

Note: Default paths are used below and may differ from your configuration.

Note: Whenever you edit the Oracle Transportation Management property files, only edit the `glog.properties` file. Any changes that you make to any other files will be overwritten every time an update is applied.

General Instructions on Upgrading Oracle Transportation Management from One Release to Another

Unless otherwise specified in release-specific instructions, please follow these steps when upgrading to a new Oracle Transportation Management version:

1. Upgrade and migrate your data to the latest roll-up (RU) of your current version of Oracle Transportation Management.
2. If not already on version 6.2.x of Oracle Transportation Management:
 - a. Upgrade to Oracle Transportation Management version 6.2.
 - b. Upgrade to the latest RU for version 6.2.
3. Install the new Oracle Transportation Management version.
4. Install the latest RU of the new version.
5. Migrate your data from the old version to the new version according to the new version's Migration Guide.
6. Run any RU-specific instructions that come out of step #4.
7. If your Oracle Transportation Management instance has any of the customizations listed below, you will need to do the following on every application and web server:
 - a. Run the customization migration creation script (see the Oracle Transportation Management Install Guide for more details).
 - b. Inspect the resulting script, modifying it for your specific instance as needed.
 - c. Run the script to do the actual migrations

This will ensure that you have the most up-to-date copies of the migration scripts in place before running the migration itself.

Note: These steps should be executed on a test instance that mimics your production instance before running them in the production instance itself. This will allow you to identify any issues and construct a solution for them before affecting your production environment.

Migration example:

- Current Version: 5.5.05.03
- Target Version: 6.3.2

Steps

1. Install and migrate to the 5.5.05.07
2. Install 6.2
3. Install 6.2.8
4. Run any 6.2.8-specific instructions
5. Install 6.3
6. Install 6.3.2

7. Run 6.3 migration scripts according to the latest Migration Guide
8. Run any 6.3.2-specific instructions
9. Create, inspect and run the customizations migration script

Branding

Oracle Transportation Management upgrades typically include updated web files (XSL, html, jpg, gif, JS, etc.) to fix specific reported problems and to provide minor enhancements requested by customers.

Since each Oracle Transportation Management upgrade includes a list of updated files, you should apply the upgrade to your test site, re-brand the updated files, and then test everything. Once it has passed your testing, you should apply the upgrade and your new branded files to your production site. This procedure ensures that you have all of the fixes included in the latest upgrade and that your branding is not broken on your production site.

Customizations

The Oracle Transportation Management 6.3 upgrade will include a step that migrates customizations to a new directory hierarchy in order to isolate these files and make future migrations easier.. The sections below list all of the files that will be migrated in this manner. Note that many of the directory names below are governed by specific, obscure properties; the actual property value will be used in the migration script creation, and are left out here for clarity purposes. Please contact Oracle Support for more information if you believe that the migration of your customizations is not being handled correctly.

Note: Because of the new hierarchy, the relative path of file includes in some customized XSL files may have broken. Carefully review your customized files and test your branding changes out fully in a test environment before moving these changes onto a production system.

Note: The directories in the following section are controlled by the `glog.customPath` property. The directory defined by that property is represented by "custom" in the paths listed in the following seven subsections.

Custom Text Emails & Faxes

As part of the 6.3 installation, customized text email and fax files will be moved from this directory:

```
<otm_install_dir>/apache/htdocs/xsl/custom
```

to this directory:

```
<otm_install_dir>/glog/gc3webapp/WEB-INF/xsl/custom
```

Also, customized text email and fax files will be moved from this directory:

```
<otm_install_dir>/apache/htdocs/xsl/<pref>
```

to this directory:

```
<otm_install_dir>/glog/gc3webapp/WEB-INF/xsl/custom/<pref>
```

where `<pref>` is a series of values selected out of the `CONTACT_PREFERENCE` table using the following SQL:

```
SELECT distinct(value) FROM contact_preference WHERE preference_gid =  
'STYLESHEET';
```


Note that any preference directory not referenced explicitly in this manner will not be migrated and should be migrated by hand.

UI Branding – XSL Files

As part of the 6.3 installation, customized UI XSL files will be moved from this directory:

```
<otm_install_dir>/apache/htdocs/xsl/custom
```

to this directory:

```
<otm_install_dir>/glog/gc3webapp/WEB-INF/xsl/custom
```

Also, customized UI XSL files will be moved from this directory:

```
<otm_install_dir>/apache/htdocs/xsl/<pref>
```

to this directory:

```
<otm_install_dir>/glog/gc3webapp/WEB-INF/xsl/custom/<pref>
```

where <pref> is a series of values selected out of the USER_PREFERENCE_D table using the following SQL:

```
SELECT distinct(user_preference_value) FROM user_preference_d WHERE  
preference_gid = 'STYLESHEET';
```

Note that any preference directory not referenced explicitly in this manner will not be migrated and should be migrated by hand.

UI Branding – Images

As part of the 6.3 installation, customized UI image files will be moved from this directory:

```
<otm_install_dir>/apache/htdocs/images/themes/<pref>
```

to this directory:

```
<otm_install_dir>/web/htdocs/images/custom/<pref>
```

and from this directory:

```
<otm_install_dir>/apache/htdocs/images/<lang>/themes/<pref>
```

to this directory:

```
<otm_install_dir>/web/htdocs/images/custom/<pref>/<lang>
```

where <pref> is a series of values selected out of the USER_PREFERENCE_D table using the following SQL:

```
SELECT distinct(user_preference_value) FROM user_preference_d WHERE  
preference_gid = 'STYLESHEET';
```

and <lang> is a series of values selected out of the TRANSLATION_D table use the following SQL:

```
SELECT distinct(text) FROM translation_d WHERE translation_gid =  
'images.root_dir';
```

Note that any preference or language directory not referenced explicitly in this manner will not be migrated and should be migrated by hand.

UI Branding – CSS Files

As part of the 6.3 installation, customized UI CSS files will be moved from this directory:

```
<otm_install_dir>/apache/htdocs/css/themes/<pref>
```

to this directory:

```
<otm_install_dir>/web/htdocs/css/custom/<pref>
```

where <pref> is a series of values selected out of the USER_PREFERENCE_D table using the following SQL:

```
SELECT distinct(user_preference_value) FROM user_preference_d WHERE  
preference_gid = 'STYLESHEET';
```

Note that any preference directory not referenced explicitly in this manner will not be migrated and should be migrated by hand.

UI Branding – JS Files

As part of the 6.3 installation, customized UI JS files will be moved from this directory:

```
<otm_install_dir>/apache/htdocs/js/<pref>
```

to this directory:

```
<otm_install_dir>/web/htdocs/js/custom/<pref>
```

where <pref> is a series of values selected out of the USER_PREFERENCE_D table using the following SQL:

```
SELECT distinct(user_preference_value) FROM user_preference_d WHERE  
preference_gid = 'STYLESHEET';
```

Note that any preference directory not referenced explicitly in this manner will not be migrated and should be migrated by hand.

Custom Engines

As part of the 6.3 installation, custom engine files will be moved from this directory:

```
<otm_install_dir>/glog/gc3webapp/WEB-INF/lib/external
```

to this directory:

```
<otm_install_dir>/glog/gc3webapp/WEB-INF/lib/external/custom
```

Integration XSL Files

As part of the 6.3 installation, integration XSL files will be moved from this directory:

```
<otm_install_dir>/glog/integration/client/xsl
```

to this directory:

```
<otm_install_dir>/glog/integration/client/xsl/custom
```

Changing Logging Options

This information has moved to the Oracle Transportation Management Security Guide.

Changing Notification Settings

The Oracle Transportation Management server sends out a variety of notifications to users. Changing these settings involves modifying the `glog.properties` file on your Oracle Transportation Management application server, or modifying the `APP_CUSTOM` property set, as described below.

```
glog.mail.smtp.host=smtp.company.com
```

This setting defines the SMTP server that Oracle Transportation Management uses when sending email and fax notifications.

```
glog.workflow.notify.advisor.email=OTMAdvisor@company.com
```

This setting defines the email address that email and fax notifications will appear to come from. This email address should be valid and this email box should be monitored, so that bounced emails and delivery failures are caught. FAXmaker will also send fax success/failure responses to this email address.

```
glog.workflow.notify.advisor.fax=800-555-1212
```

This setting defines the fax number that fax notifications appear to come from.

```
glog.workflow.notify.FAXmaker.email=fax@company.com
```

This setting defines the e-mail address of the fax server. The fax server should monitor the corresponding mailbox and convert the incoming emails to faxes.

Multi-byte Character Support in BI Publisher

Oracle BI Publisher PDF generation with multi-byte characters requires an extra font be copied into a specific directory. Please follow these instructions when Oracle Transportation Management is first installed, and every time the JDK is updated.

1. Log into each server where report transformation is being done.
2. Copy the files in the `<otm_install_path>/install/fonts` directory to the `<otm_install_path>/jdk/jre/lib/fonts` directory:

```
cp <otm_install_path>/install/fonts  
<otm_install_path>/jdk/jre/lib/fonts
```
3. Restart the Oracle Transportation Management instance.

Units of Measure

Oracle Transportation Management stores all amounts in two units of measure: the actual unit of measure and a storage default unit of measure. The storage default amount is stored in the database "BASE" columns and is used to support querying amounts (i.e. Shipment Total Weight) which have different actual units of measure. The storage default is designated by an indicator on the unit of measure for each unit of measure type (weight, volume, distance, etc.). The default for storage default uses U.S. standard units of measure.

In most cases, the unit of measure displayed on the user interface is controlled by a user preference. If a user does not have a user preference defined, the unit of measure is controlled by the Display Default indicator on the unit of measure for each unit of measure type. The default for Display Default uses U.S. standard units of measure.

The Display Default and Storage Default settings can be modified by running an action on the corresponding unit of measure. In addition, new units of measure and corresponding conversions can be created. The Unit of Measure page is located at **Configuration and Administration > Power**

Data > General > Unit of Measure. This page is only available when you are logged in as DBA.ADMIN. For more information on this topic refer to the help topic "Unit of Measure".

Changing Currency Settings

By default, Oracle Transportation Management uses US Dollars when saving costs to the database. Also by default, Oracle Transportation Management triangulates all currency conversions through US Dollars.

This example illustrates how Oracle Transportation Management stores a shipment cost record with the currency storage default set to two different currencies:

Total actual cost of shipment is 1000 JPY. If Oracle Transportation Management's currency storage default is USD (current default in all Oracle Transportation Management installations), Oracle Transportation Management stores this cost as follows:

Total_actual-cost	Total-actual-cost_currency_GID	Total_Actual_cost_base
1000	JPY	7.76

If instead Oracle Transportation Management's currency storage default is GBP:

Total_actual-cost	Total-actual-cost_currency_GID	Total_Actual_cost_base
1000	JPY	5.31

In the first instance, the rate of 7.76 represents the USD value of 1000 JPY converted at the current rate in Oracle Transportation Management (128.77) while in the second instance, the rate of 5.31 represents the GBP value of 1000 JPY converted at the current JPY/GBP rate in the system (188.08).

When to Change Currency Storage Default

There are two scenarios where you would like to change your currency storage default: either you only use one currency other than USD or you use multiple currencies but not USD.

Rates in One Single Currency

In this case, you only need to set your currency storage default to the currency you use. For example, Oracle Transportation Management stores a 100 GBP shipment cost as 100 in both the total cost and the total cost base fields so no currency conversion is needed.

1. Run the following insert statement (example using GBP as new storage default) as the DBA user:

```
INSERT INTO UOM (TYPE, UOM_CODE, UOM_SHORT_DESCRIPTION,  
UOM_LONG_DESCRIPTION, UOM_RANK, IS_STORAGE_DEFAULT,  
IS_DISPLAY_DEFAULT, DOMAIN_NAME) VALUES ('CURRENCY', 'GBP', 'UK  
POUND', 'UK POUND', 1, 'Y', 'Y', 'DBA')
```

Rates in Several Currencies

In this case, you need to:

1. Decide what the currency storage default is for your Oracle Transportation Management installation.

2. Set your currency storage default by running the following insert statement (example using GBP as new storage default) as the DBA user:

```
INSERT INTO UOM (TYPE, UOM_CODE, UOM_SHORT_DESCRIPTION,  
UOM_LONG_DESCRIPTION, UOM_RANK, IS_STORAGE_DEFAULT,  
IS_DISPLAY_DEFAULT, DOMAIN_NAME) VALUES ('CURRENCY', 'GBP', 'UK  
POUND', 'UK POUND', 1, 'Y', 'Y', 'DBA')
```

Oracle Transportation Management still needs currency rates to convert between the currencies you use. You can download rates from the IMF website. (This populates the DEFAULT rate in the CURRENCY_EXCHANGE_RATE table.) The problem is that all rates from the IMF are against USD. You have two alternatives to using USD centric IMF rates:

- Write an API that uses another source than the IMF to bring in exchange rates stated against your currency storage default.

or

Update the rates you need manually or use a CSV process while again entering your needed rates against your preferred currency. You can do this nightly, monthly or in any other frequency.

- In the `glog.properties` property file or the `CUSTOM` property set, set `glog.currency.base` to your currency storage default (e.g. EUR). This makes Oracle Transportation Management triangulate through the currency of your choice.

This means that Oracle Transportation Management will have all DEFAULT rates stated against your base currency and triangulates using your base currency.

Accounting for Missing Time Zones

The UTC pl/sql package depends on java running in Oracle to do time zone conversions. Some time zones are not accounted for in the `java.util.TimeZone` class. The solution to this problem involves a new (and optional) "missing_time_zone" table, which has the following structure:

Column	Value
TIME_ZONE_GID	NOT NULL VARCHAR2(128)
ALT_TIME_ZONE_GID	NOT NULL VARCHAR2(128)
ADJ_FACTOR	NOT NULL NUMBER

So for example, if you had a location that specified a time zone of "FOO", and you wanted "FOO" to be treated the same as "America/New_York", you would specify a missing_time_zone record of:

```
time_zone_gid = 'FOO'  
alt_time_zone_gid = 'America/New_York'  
adj_factor = 0
```

The `adj_factor` is added to the `time_zone_offset`. So if the `time_zone_offset` for America/New_York was -5, and you wanted 'FOO' to have an offset of -4, you could set the `adj_factor` to 1.

Some of the missing Australian time zones are accounted for in the java code itself. The defaults established in the code for the Australian time zones can be overridden via the missing_time_zone table.

Country Code Conversion from ISO Alpha-3 Code to Alpha-2 Code

1. Oracle Transportation Management uses ISO alpha-3 code format for the country code. Country code format can be change to ISO alpha-2 code as described below. This process will change 3 character country code to 2 character country code in the Oracle Transportation Management database. Prior to running this process, ensure that the two character country code record exists on the Country Code table for the country codes which were not supplied by Oracle Transportation Management. Go to <glog_home>/glog/oracle/script8 directory of the application server
2. Log into database as user *GLOGOWNER* using SQL PLUS and run following SQL command:
`@convert_cc3_cc2.sql`
3. Review and resolve any errors shown in the log file `convert_cc3_cc2.log`.

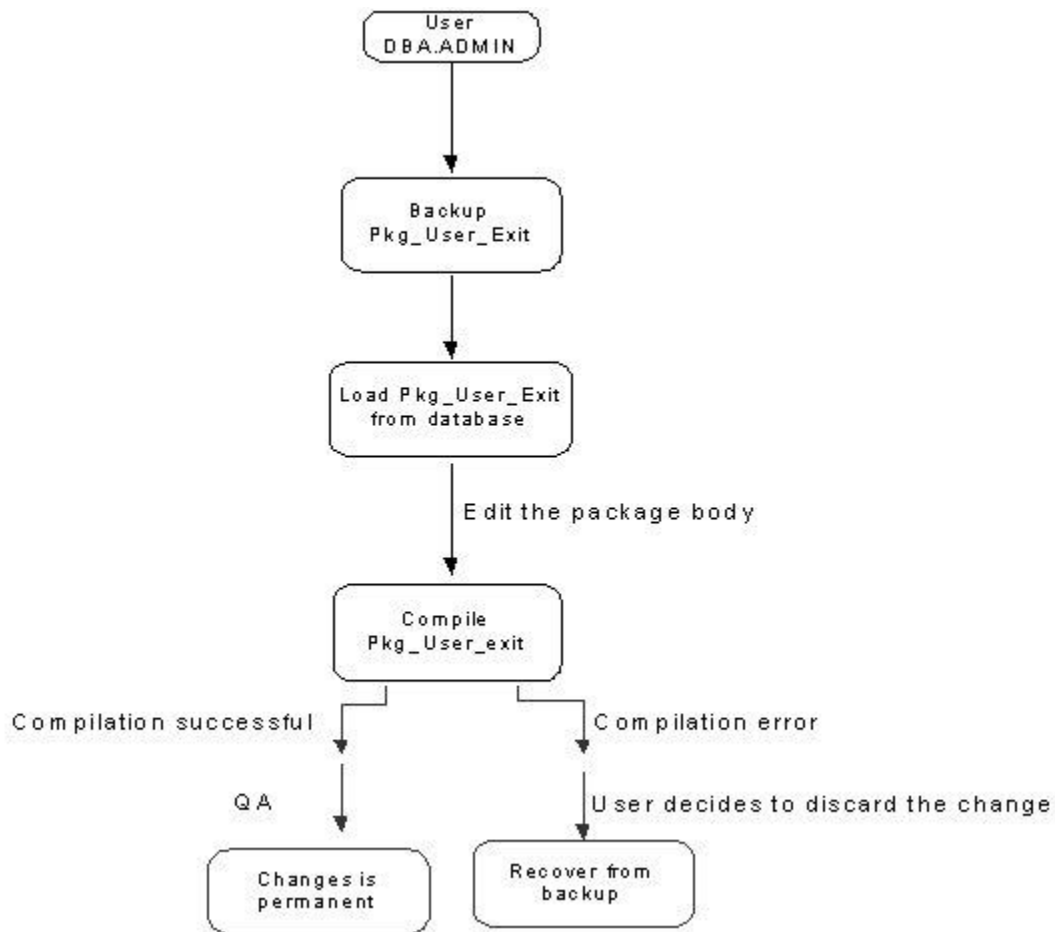
Data Validation

Data Validation allows you to alter default user exit validation behavior through Oracle Transportation Management. The user exit validation is performed through `pkg_user_exit` package. This increases the transparency of the user exit validation. You are able to alter and compile the package.

A validation package is provided, `pkg_user_exit`, which only serves as a template. It is called by persistence engine and performs the post-persistence validation when the property parameter `glog.persistence.callout` is set to On. By default, the package does not do anything when executed. It has only one public stored procedure: `to_exit`.

The stored procedure delegates the validation works to other private stored procedures, which should be developed by users according to their business requirements. However, the public stored procedure in package spec should never be modified.

Process Flow



APIs for Compiling `pkg_user_exit`

UserExitCompiler Class

The `UserExitCompiler` should be able to load `pkg_user_exit` package from database and compile it to the database. Additionally, it can perform backup and recovery of the package. The backup and recovery allows you to restore the package to previously functional state when the edited package compilation is failed or the user decides to discard the altered package. To avoid security related issues, this class can only modify `pkg_user_exit` package, which is only restricted to package body. Furthermore, only the user `DBA.ADMIN` is allowed to edit the package. In the class, the following public methods will be included. The return of the methods is either status of the action or error messages when the database action is failed.

- `public String get ()` - Load the package from database.
- `public String getBackup ()` - Load backup copy from database.
- `public String Backup ()`.
- `public String Restore ()`.
- `public String put (String packageString)` - compile the package into database .

Pkg_SQL_executor Package

Class UserExitCompiler connects database through GLOGDBA. However, GLOGDBA does not have privileges to create database objects in Glogowner schema. To circumvent the issue, a new package called Pkg_SQL_executor is created to compile altered pkg_user_exit package in Glogowner schema. The privilege to execute the package is granted to GLOGDBA. The package is executed from UserExitCompiler.

HTTP Integration

Oracle Transportation Management uses HTTP to communicate with a number of downstream systems. Canonical XML data is packaged into an HTTP request and processed asynchronously through a set of workflow threads. If the downstream system is slow to respond, these threads can backup and impact system performance.

By default, the system waits indefinitely for a downstream system response. This is to minimize the number of failed XML transmissions requiring reprocessing. Timeouts, however, can be specified in two ways:

- If the transmission is directed via an External System specification, the connection and read timeouts are given by the **Socket Timeout** on the External System. If 0, the system waits indefinitely.
- If the transmission is specified with an ad-hoc URL, the following properties control timeout:
`glog.httppost.connectTimeout=<timeout in seconds>`
`glog.httppost.readTimeout=<timeout in seconds>`

To have the properties take precedence over the External System timeouts, set:

```
glog.httppost.allowPerSystemTimeouts=false
```

BPEL Integration

BPEL Flows

Oracle Transportation Management provides BPEL integrations with Oracle E-Business Suite. These flows can be found in `<otm_install_path>/utils/integration/bpel/flows`. Each flow should have a README.TXT file in the zip file with further instructions. There are several other flows that ship with the other Oracle products. Refer to the respective product documentation for the details for those flows.

Oracle Transportation Management Properties Files for Oracle E-Business Suite Integration through BPEL

If integrating with Oracle E-Business Suite through Oracle BPEL Process Manager, you need to include/uncomment the ebs.properties file in the glog.properties file on the Oracle Transportation Management web and application servers. These files are usually under `<otm_install_path>\glog\config` on Windows or `<otm_install_path>/glog/config` on UNIX. In the `glog.properties` file, you would uncomment the following⁴:

```
#!/include ebs.properties
```

⁴ Note that this must be done in the `glog.properties` file. There is no method to include a property file from within the `CUSTOM` property Set.

Your line would look like this:

```
!include ebs.properties
```

Once this change has been made, restart your Oracle Transportation Management instance.

Oracle Transportation Management Properties Files for BPEL Integration

If integrating with the Oracle BPEL Process Manager, but not necessarily integration with Oracle E-Business Suite, you will need to enable a property on the Oracle Transportation Management web and application servers. In the `glog.properties` file or the `CUSTOM` property set, add the following:

```
glog.integration.enableParserInServlets=true
```

Once this change has been made, restart your Oracle Transportation Management instance.

Recurring Processes

This information has been moved to the Oracle Transportation Management Security Guide.

SMTP Integration

Oracle Transportation Management uses SMTP to communicate with an external mail server. Mail requests are routed to a set of dedicated threads that processes the SMTP request and response. If the mail server is slow to respond, these threads can backup and impact system performance.

By default, the system waits indefinitely for a mail server response. This is to minimize the number of failed e-mails. Timeouts, however, can be specified by setting the following properties:

```
glog.mail.smtp.readTimeout=<timeout in milliseconds>
glog.mail.smtp.writeTimeout=<timeout in milliseconds>
glog.mail.smtp.connectTimeout=<timeout in milliseconds>
```

Note that this section used to hold information regarding SMTP authentication. This has been moved to the Oracle Transportation Management Security Guide.

11. Advanced Configuration: LDAP

This information has been moved to the Oracle Transportation Management Security Guide.

12. Performance Monitoring

Performance monitoring is a necessary part of running a complex multi-tier application like Oracle Transportation Management. In most cases, a sudden change in performance is caused by a recent configuration change or an increase in transactional data. A formal change management process is critical for maintaining a stable production system and for quickly diagnosing performance issues should they happen. All system related configuration changes should be tested before applying to production. The date and time of the change should also be tracked. In a situation where this process is not being followed, a query of the insert/update date on important configuration tables may reveal a recent change that is causing an issue. It is also important to maintain benchmark statistics (i.e. orders and shipments processed in a typical day) to be able to quickly rule out a significant increase in data volume.

Operating System Monitoring Tools

Microsoft Windows

On Microsoft Windows, you can monitor system performance using the Task Manager or Performance Monitor. You should monitor general trends, such as CPU usage, memory usage, and memory swapping (out to virtual memory).

The Oracle HTTP Server (OHS) service starts up multiple instances of OHS to handle incoming requests. Each of these will show up as "Apache.exe", and can be monitored for performance statistics using general Microsoft Windows Server tools.

The Tomcat service starts up one Java Virtual Machine (JVM) and shows up as "java.exe". Unfortunately, this can be hard to find if you have multiple JVMs running on the same machine. Usually you can identify Tomcat by the amount of memory it is using, as normal Tomcat instances use large amounts of memory (1.6 GB or higher). You can monitor the performance statistics of Tomcat using general Microsoft Windows Server tools.

The WebLogic service starts up one JVM, but shows up as beasvc.exe, because of WebLogic's special service runner. You can monitor the performance statistics of WebLogic using general Microsoft Windows Server tools. You can also get detailed performance statistics from the WebLogic console. Follow these steps:

1. Log into the WebLogic console (http://<appserver_name>:7001/console) as the user system.
2. In the left frame, expand Otmv63 > Environment > Servers.
3. Select the Oracle Transportation Management server (e.g. gc3-otm-machinename-63-wl).
4. In the main frame, select the Monitoring tab.
5. Then select the Performance tab.
6. This screen shows you the following:
 - Request Throughput: The rate at which requests are processed. The higher the better.
 - Requests Waiting: The number of requests waiting to process. The lower the better.
 - Memory Usage: The amount of memory utilized by WebLogic. Keep in mind that this shows you how much memory WebLogic is using, within the memory allocated to the JVM. The amount of memory reported by your operating system will be higher, since the JVM reserves all of the memory that it has been configured to allocate, regardless of whether WebLogic is using it all or not. This was set during the Oracle Transportation Management installation. To get a true memory reading, you should force a garbage collection.

- Force Garbage Collection: This will force the JVM to free up all unused memory. This should be run in order to get a true memory reading from the main performance screen. Keep in mind that forcing a garbage collection will slow the server down and shouldn't be forced continually. The JVM usually manages garbage collections automatically as the server runs.
7. To monitor database connections, expand Services > JDBC > Connection Pools.
 8. Select dbaPool.
 9. Select the Monitoring tab and Monitor all instances of dbaPool.
 10. This page shows you the initial, maximum, and high number of db connections. You should monitor whether the high number is at or near the maximum number of connections. If so, you should consider increasing the database connections in the dbaPool.

UNIX

On UNIX, you can monitor system performance using the ps, top, or sar. You should monitor general trends, such as CPU usage, memory usage, and memory swapping (out to virtual memory). All processes for Oracle Transportation Management should run under the user setup during the installation process. This can be helpful for identifying the processes.

The Oracle HTTP Server (OHS) service starts up multiple instances of OHS to handle incoming requests. Each of these will show up as "httpd" or "httpd.worker", and can be monitored for performance statistics using general UNIX tools.

Tomcat

The Tomcat service starts up one JVM and shows up as "<path>/ java -server -Xms1600m -". Unfortunately this can be hard to find if you have multiple JVMs running on the same machine. Usually you can identify Tomcat by the amount of memory it is using, as normal Tomcat instances use large amounts of memory (1.6GB or more). You can monitor the performance statistics of Tomcat using general UNIX tools.

WebLogic

The WebLogic service starts up one JVM and shows up as "<path>/ java -server -XX:MaxPer" (differs slightly from the Tomcat instance). Unfortunately this can be hard to find if you have multiple JVMs running on the same machine. Usually you can identify WebLogic by the amount of memory it is using, as normal WebLogic instances use large amounts of memory (1.6 GB or more). You can monitor the performance statistics of WebLogic using general UNIX tools. You can also get detailed performance statistics from the WebLogic console. Follow the steps under the **Microsoft Windows** section in the **Operating System Monitoring Tools** section.

SAR

The "sar" command-line utility displays CPU activity. Running "sar 10 10" will run the sar command ten times in ten second intervals. The output will show basic CPU, %nice, %system, %iowait and % idle information.

```
Linux 2.4.21-37.0.1.ELsmp (spapp01)      01/30/2008

07:24:18 AM      CPU      %user      %nice      %system      %iowait      %idle
07:24:28 AM      all        2.88        0.00        0.28        0.00        96.85
07:24:38 AM      all        3.65        0.00        0.60        0.10        95.65
```

Oracle Transportation Management activity will show up under the %user column, high activity in the user column would be expected on an unresponsive system. If you see significant utilization in the

system or iowait columns it could be an indication of other internal processes or swapping memory to and from disk that may need to be investigated further.

TOP

The “top” command-line utility produces a frequently-updated list of processes on most UNIX-based platforms. By default, the processes are ordered by percentage of CPU usage, with only the highest CPU consumers shown. The top command shows how much processing power and memory are being used. Running the “top” command will give you information on the system memory and swap space usage. The output on the screen is broken down into two sections, the top section tells you the current system time, up time, number of users and average load.

```
top - 08:57:58 up 97 days, 23:53, 3 users, load average: 0.06, 0.03,
0.00
Tasks: 79 total, 1 running, 78 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0% us, 0.0% sy, 0.0% ni, 100.0% id, 0.0% wa, 0.0% hi, 0.0% si
Mem: 2049976k total, 2034984k used, 14992k free, 1848k buffers
Swap: 2048248k total, 1021960k used, 1026288k free, 14252k cached
```

It also shows you the total amount of memory available on the system, amount used and free. The same information is shown for the swap space allocated on the system, swap used, swap free and swap cached.

The bottom section of the output shows the top 20 processes listed by PID and user name, priority and other information. For many Oracle Transportation Management instances Java will likely show up as being the highest consumer of memory and CPU on the system.

PID	USER	PR	NI	VIRT	RES	SHR S	%CPU	%MEM	TIME+	COMMAND
1	root	16	0	4752	76	48 S	0.0	0.0	0:01.93	init
2	root	RT	0	0	0	0 S	0.0	0.0	0:01.84	migration/0
3	root	34	19	0	0	0 S	0.0	0.0	0:00.04	ksoftirqd/0
4	root	RT	0	0	0	0 S	0.0	0.0	0:01.86	migration/1
5	root	34	19	0	0	0 S	0.0	0.0	0:00.03	ksoftirqd/1
6	root	5	-10	0	0	0 S	0.0	0.0	0:00.01	events/0
7	root	5	-10	0	0	0 S	0.0	0.0	0:00.01	events/1
8	root	5	-10	0	0	0 S	0.0	0.0	0:00.00	khelper
9	root	13	-10	0	0	0 S	0.0	0.0	0:00.00	kacpid
33	root	5	-10	0	0	0 S	0.0	0.0	0:00.00	kblockd/0

Note: If you see other processes that are using significant system resources you should ask the administrator to follow up.

Oracle Transportation Management Tools

Configuration Data Collection

The Configuration Data Collection tool captures important configuration data that can be used in diagnosing a performance issue. Users have the option of displaying the information on the screen, or they can have the data collected to them via FTP or via email. Information sent via FTP and email is encrypted, but this can be disabled via the following properties:

```
glog.diag.output.ftp.encrypt=false
glog.diag.output.email.encrypt=false
```

Files that are not encrypted and sent via email are given the .ZOP extension.

In addition to this data, there is also an option to Export the Environment. This will send you the most recent access log from Oracle HTTP Server (OHS), config.xml.original from weblogic, console.log from Tomcat and WebLogic, Error.log from the Oracle Transportation Management logs directory, glog.exception.log, glog.properties, httpd.conf, mod_jk.log and mod_jk.conf from apache, tomcat.conf, weblogic.conf, weblogic.sh, workers.properties, several class files and the patch notes from the installed releases on the system. The list of files to be exported is stored in a new properties file called `glog.environment.export.properties`. You can also add files to the list on the fly by adding an additional `glog.environment.export` entry via the properties servlet.

Performance Data Collection

The Performance Data Collection utility can be used to capture performance statistics that can be used in diagnosing a performance issue. Users have the options of displaying the information on the screen or having it sent via FTP or email to a contact they have defined in the system.

Web Server Tools

Load Balancers

A test servlet is provided to assist in determining whether or not a web server should be dropped from or restored to a load balancer rotation. Please see the *Installing Multiple Web Servers* section above for more details.

Application Server Tools

Administration Console

The application servers used by Oracle Transportation Management have administration consoles that provide various monitoring utilities. This document will not cover the monitoring utilities, where they are, or how to use them. Please see your application server documentation for the specific details on accessing the console and how to use the utilities that are provided. However, there are certain tips that we will provide.

Monitoring Deployed Enterprise Java Beans

The administration console for the application server should have the ability to monitor Enterprise Java Beans (EJB). Oracle Transportation Management only uses stateless Enterprise Java Beans so keep this in mind when using the administration console to monitor these. Monitoring really only needs to be for heavily used EJBs like `workflow.WorkflowSession`.

JVM Tools

Console Logs

For the web and application servers, system monitoring utilities will only be partially effective since the memory of a Java application is typically static and limited by the Java heap size. They will be useful if multiple applications are being run on a single machine and the total physical memory may be limited. If the entire Java heap cannot be contained in physical memory performance issues will be very likely due to memory swapping during JVM garbage collection.

Memory monitoring within the Java heap can be performed by various Java monitoring tools or by reviewing the verbose garbage collection output in the console.log files. You should have garbage collection logging enable. This is done via one of the parameter settings in the ".conf" file for Tomcat, WebLogic. Console logs for the web server and the application server should be reviewed to understand the memory allocation and garbage collection.

Thread Dumps

Thread dumps are critical when diagnosing almost any performance issue. They are particularly useful when diagnosing an issue where the application server is unresponsive. Thread dumps capture the current state (executing, blocked, or inactive) and a stacktrace indicating the exact code that is being executed for every thread. Thread dumps should be taken in a set of three thread dumps taken consecutively at 30 second intervals. If the performance issue continues for an extended period of time, it is extremely useful to capture multiple sets of thread dumps.

If there is very little activity in the thread dumps the cause of the performance issue may be fairly apparent. However, if there are many active threads it may be very difficult to deduce the exact cause of the performance issue. In this case, capturing the thread dumps during multiple occurrences of the issue is necessary.

Heap Dumps

A heap dump is a dump of all the objects that are currently in the Java heap. The heap dump is a critical piece of information when diagnosing a memory related issues. The heap dump can be captured and then be analyzed off-line. Unfortunately the support and technique for capturing a heap dump varies by JVM and version of the JVM. Oracle Support will help with capturing a heap dump when it is necessary.

Capturing JRockit Mission Control JRA Files

Mission Control is a run-time analysis tool that is included with the JRockit JVM. This utility captures and analyzes JRA (Java Run-time Analysis) files. This utility can be used to analyze performance, memory heap usage, garbage collection, and thread latency. Mission Control imposes minimal overhead on the running system and therefore can be used to analyze issues in a production environment. If the issue you are experiencing only occurs sporadically it can be helpful to capture JRAs both prior to the problem happening and while it is happening for comparison purposes. The following steps can be used to capture a JRA.

1. Find the Process Id (PID) of the application server and keep it for the next step. You can do this by running this command: `<OTM_INSTALL_DIR>/jdk/bin/jrcmd -p`
 - The `<OTM_INSTALL_DIR>/jdk/bin/jrcmd -p` command will list all of the java processes running.
2. You can then make a jra recording from command line by running this command after replacing the information in between the :

```
<OTM_INSTALL_DIR>/jdk/bin/jrcmd <PID> jrarecording time=300  
filename=<OTM_INSTALL_DIR>/logs/perfIssue.xml.zip  
threaddumpinterval=30s
```

Please note the time is in seconds and this can be modified to any number of seconds. The directory and filename can also be changed.

The JRA utility will write to the file for the entire time interval specified. Every time you run it, you would probably want a new filename to help differentiate between the different records. You can always just save the file off somewhere else and rename it. JRA will record until the time interval is over, and then stop. There is no manual intervention needed to turn off the profiler (JRA recording).

Database Monitoring Tools

Oracle Enterprise Manager includes several tools which are critical to monitoring the performance of the Oracle Database. Two of these tools are Automatic Workload Repository (AWR) and Automatic Database Diagnostic Monitor (ADDM), both of which are included in the Diagnostics Pack.

Automatic Workload Repository

By default snapshots of the relevant data are taken every hour and retained for seven days. The data is accessed via a report by running `awrrpt.sql` or `awrrpti.sql`. The automated workload repository administration tasks have been included in Enterprise Manager. The Automatic Workload Repository page is accessed from the main page via Administration > Workload Repository (under the Workload section).

Automatic Database Diagnostic Monitor

ADDM analyzes data in the Automatic Workload Repository to identify potential performance bottlenecks. For each of the identified issues it locates the root cause and provides recommendations for correcting the problem. An ADDM analysis task is performed and its findings and recommendations stored in the database every time an AWR snapshot is taken.

The best place to start viewing ADDM reports is Enterprise Manager. The Performance Analysis section on the Home page is a list of the top five findings from the last ADDM analysis task. Specific reports can be produced by clicking on the Advisor Central link, then the ADDM link. The resulting page allows you to select a start and end snapshot, create an ADDM task and display the resulting report by clicking on a few links. Alternatively the `addmrpt.sql` script can be used to create an ADDM report from SQL*Plus.

Network Monitoring

Due to the heavy traffic between the client, web, application, and database, a small change in network latency can have a significant impact on system performance. System monitoring tools will be helpful in diagnosing network performance issues. Generally speaking though, network performance issues are one of the most difficult to identify. Maintaining benchmarks of good network performance that can be referenced during performance issues is a good practice.

Diagnosing a Performance Issue

Steps for Diagnosing a Performance Issue

1. Determine the scope of the problem.
 - Is this issue affecting one user or many users?
 - Can new users login to Oracle Transportation Management?

- Are agents and integrations not processing?
 - Record the exact timeframe of when the issue occurred.
2. Monitor CPU, I/O, and network activity.
Use a system monitoring tool to capture and review whether there is abnormally high CPU, network, or I/O activity on the web/app/database.
 3. Check the exception logs.
Capture and review the exception.log file for unusual exceptions.
 4. Review logging that is enabled.
Review what logging is enabled via Logging Overview page and potentially suppress all logging.
 5. Review workflow activity.
Capture and review Event Diagnostics servlet or Event Queue Application Performance Diagnostic for a backlog of unprocessed events.
 6. Review console.log files.
Capture and review the console.log file looking for excessive garbage collection activity.
 7. Capture thread dumps.
Perform multiple sets of three thread dumps (30 seconds apart) on the web/applications servers.
 8. Restart the server.
If the problem persists, a restart may be necessary to resolve the issue.
 9. Review change log for recent configuration changes.
Most performance problems are caused by a recent configuration change. Review the log of recent changes and provide this information to Support.

Additional steps for Database Related Issue

10. Capture AWR/ADDM reports.
11. Review reports for SQL which may need tuning.
12. Check for stale and partial database statistics.

13. Performance Tuning

JVM

The JVMs supported by Oracle Transportation Management vary by platform. Each JVM has a set of parameters which can be used to tune the performance of the JVM. Most of these parameters affect the memory management process referred to as garbage collection. Oracle Transportation Management ships with a default set of these parameters to be used as a starting point for JVM tuning. Changing these parameters can have a significant impact on the overall performance of the system. Therefore, changes should be well documented and made incrementally with thorough performance testing.

Changing Memory Settings

As your server load varies, you may need to change the amount of memory that is allocated to the Oracle Transportation Management instance. You can change the amount of memory allocated to both Tomcat and WebLogic.

Tomcat is the Java servlet server that Oracle Transportation Management uses. Increasing the amount of memory that it can utilize allows more Oracle Transportation Management concurrent users and improves the speed of some integration tasks.

WebLogic is the application server that Oracle Transportation Management runs within. Increasing the amount of memory that it has improves performance of business logic functions.

After making any memory changes, you need to stop and restart Oracle Transportation Management, as described starting on page in the chapter **Starting and Stopping the Oracle Transportation Management Servers**. The instructions below describe how to make these changes.

UNIX and Microsoft Windows

To change the amount of memory that Tomcat uses, edit `<otm_install_path>/tomcat/bin/tomcat.conf` and modify the following lines:

```
jvm.arg=-Xms[TOMCAT_MEMORY]m
jvm.arg=-Xmx[TOMCAT_MEMORY]m
```

[TOMCAT_MEMORY] will be some integer value that represents the amount of memory (in megabytes) to use. For instance, if you changed this to 2.2GB, the line would read:

```
jvm.arg=-Xms2200m
jvm.arg=-Xmx2200m
```

To change the amount of memory that WebLogic uses edit `<otm_install_path>/weblogic/weblogic.conf`:

```
jvm.arg=-Xms[WEBLOGIC_MEMORY]m
jvm.arg=-Xmx[WEBLOGIC_MEMORY]m
```

If you change this value to 2.2GB, it would look like:

```
jvm.arg=-Xms2200m
jvm.arg=-Xmx2200m
```

In all cases, restart your Oracle Transportation Management instance to make these changes take effect.

Web Server

Simultaneous Users

If your installation is going to have a large number of simultaneous users it may be necessary to tune some Oracle HTTP Server and Tomcat settings. It is necessary to do performance testing and monitoring in order to tune these settings for an Oracle Transportation Management installation. Depending on the volume of users and the type of activity performed by these users it may also be necessary to setup additional webserver instances.

1. `maxThreads` in `<OTM_INSTALL_DIR>/tomcat/conf/server.xml`:

The maximum number of request processing threads to be created by this connector, which therefore determines the maximum number of simultaneous requests that can be handled. If not specified, this attribute is set to 200. If an executor is associated with this connector, this attribute is ignored as the connector will execute tasks using the executor rather than an internal thread pool.

For example:

```
<Connector address="localhost" port="8009" protocol="HTTP/1.1"
           maxThreads="1000" connectionTimeout="3600000"
           URIEncoding="UTF-8" />
```

2. Change `MaxClients` in `<ohs_install_dir>/Oracle_WT1/instances/<instance name>/config/OHS/<component_name>/httpd.conf`

The following properties in the `httpd.conf` file control the number processes/threads that will be created within Tomcat. Connection attempts beyond these limits will normally be queued, up to a number based on the "ListenBacklog" directive.

For example:

- `# StartServers`: number of server processes to start
- `# MaxClients`: maximum number of server processes allowed to start
- `# MinSpareThreads`: minimum number of threads which are kept spare
- `# MaxSpareThreads`: maximum number of threads which are kept spare
- `# ThreadsPerChild`: number of threads per server process
- `# ServerLimit`: maximum number of server processes
- `# MaxRequestsPerChild`: maximum number of requests a server process serves

`<IfModule worker.c>`

```
StartServers      10
ServerLimit       40
MaxClients        1000
MinSpareThreads   25
MaxSpareThreads   75
ThreadsPerChild   25
MaxRequestsPerChild 0
```

See the documentation for Oracle HTTP Server for more information on these configuration settings.

Application Server

Logging

Although logging is a vital function in Oracle Transportation Management, excessive logging is very common cause of poor performance. This is particularly true of bulk planning processes. The Log Files and Logging Overview pages can be used to review what logs are currently enabled. You can also temporarily disable all logging by setting the following property:

```
glog.log.suppressAll=true
```

Setting this property can be a quick method of determining whether logging is the cause of a performance issue.

Note: Log File and Logging Overview pages on the Configuration and Administration menu can only be used to display and configure details about log files enabled on the application server and web server. Logging on the web server can only be controlled through `glog.properties`, the `WEB_CUSTOM` property set or Log File pages that are of type WEB.

Another way to determine if logging is impacting performance is to look at the rate at which log files are turning over. On the server running Oracle Transportation Management, you can perform a quick scan of the directory containing the Oracle Transportation Management logs files. If this reveals a log file that is turning over a 10MB log file several times per second, it is very likely to be impacting performance.

LogIds with a suffix of "Debug" or "Details" have the potential to log significant amounts of data and should be avoided unless directed to be by Support. Ad-hoc logs are the most dangerous because they generate logging regardless of the user logged in. On the other hand, User logs only write to the log file when that particular user is logged in and using Oracle Transportation Management. In some scenarios user logs can still have a significant impact on performance, even if that particular user is not logged in. This logging happens because there is a certain amount of overhead in generating a log message. The overhead occurs before Oracle Transportation Management determines, based on the logged in user, that it does not need to write the message to the log file. For this reason, having many user logs with detailed logging enabled can have a significant impact on performance.

Oracle Transportation Management Business Object Caches

The Oracle Transportation Management Business Object caches are maintained by Oracle Transportation Management. The majority of Oracle Transportation Management Business Objects caches use a Least Recently Used (LRU) strategy to maintain the cache. When an LRU cache reaches its maximum, a one-for-one exchange is made for the new object and the least recently used object in the cache. Most static data used by Oracle Transportation Management business logic is maintained in one of these caches. The App-tier Caches utility page, located on the menu under Technical Support – Diagnostics and Tools – Caches, can be used to review statistics on these caches.

The size of a Business Object Cache can have a significant impact on performance. The efficiency of a cache is measured by its hit ratio. A low hit ratio is a possible indication of an undersized cache. If a cache has reached its capacity and the hit ratio is low (less than 0.80), performance may be impacted. Increasing the maximum size of this cache may increase system performance. Temporary changes can be made to the cache using the diagnostic screen, but the changes will revert to the default upon restart. To permanently change the size of a cache the appropriate `glog.property` must be set. For example, the size of the Rate Offering cache is set by the following property:

```
glog.cache.TRateOfferingCache.capacity=2000
```

It is important to note that increasing the size of the cache has the adverse affect of increasing memory usage, so changes should be done incrementally and with thorough testing.

Business Number Generator (BNG) Contention

The Business Number Generator is an Oracle Transportation Management mechanism for creating IDs based on a complex set of business rules. Since the IDs are based on a sequence, it is necessary to generate the IDs one at a time in order to prevent duplicate IDs. For this reason, a process may have to wait for another process to finish generating an ID. This waiting can manifest itself as a performance issue. The impact is even more significant in a scalability environment since the synchronization must be coordinated across application servers.

It is a good practice to eliminate unnecessary use of BNG. For example, the Bill of Lading 'BM' Shipment reference number can be disabled with the following property:

```
glog.shipment.createBMRefnum=false
```

The Oracle Database sequence number generator has less overhead than the BNG and should be used instead of the BNG when possible. For example, the `s_ship_unit_gid` can be changed from the BNG to an Oracle Database sequence number by setting the following property:

```
glog.server.bngenerator.oracleSequence.xid.S_SHIP_UNIT_XID.DEFAULT=S_SHIP_UNIT_SEQUENCE
```

The following database sequences should be used instead of BNG in high volume transactional environment.

```
glog.server.bngenerator.oracleSequence.xid.S_SHIP_UNIT_XID.DEFAULT=S_SHIP_UNIT_SEQUENCE
glog.server.bngenerator.oracleSequence.xid.SHIPMENT_XID.DEFAULT=SHIPMENT_SEQUENCE
glog.server.bngenerator.oracleSequence.xid.ORDER_MOVEMENT_XID.DEFAULT=ORDER_MOVEMENT_SEQUENCE
glog.server.bngenerator.oracleSequence.xid.SHIP_UNIT_XID.DEFAULT=SHIP_UNIT_GID_SEQUENCE
glog.server.bngenerator.oracleSequence.xid.ORDER_RELEASE_LINE_XID.DEFAULT=ORDER_RELEASE_LINE_GID_SEQ
```

These properties follow the format shown below.

```
glog.server.bngenerator.oracleSequence.gid.<rule id>=<oracle database sequence>
```

Database sequence can be created using following steps if it is not supplied by Oracle Transportation Management.

Log into the database as glogowner using SQL PLUS and issue following SQL commands:

```
exec create_sequence.seq(P_TABLE, P_SEQUENCE, P_COLUMN);
grant select on P_SEQUENCE to APP_USER, EXT_USER;
@create_otm_synonyms.sql
```

For example, following commands will create sequence SHIP_UNIT_SEQUENCE, grant privileges, create synonyms and associate sequence SHIP_UNIT_SEQUENCE with column SHIP_UNIT_XID and table SHIP_UNIT.

```
exec create_sequence.seq('SHIP UNIT', 'SHIP UNIT SEQUENCE', 'SHIP UNIT_XID');
grant select on SHIP UNIT SEQUENCE to APP USER, EXT USER;
@create_otm_synonyms.sql
```


DirectSQLUpdate

DirectSQLUpdate is an agent action which allows you to update an attribute of an Oracle Transportation Management business object using a SQL statement. It is equivalent to a database trigger, but the agent action is capable of refreshing Oracle Transportation Management business caches to reflect the modified data. Performance tuning of SQL statements in DirectSQLUpdate is important. It is also important to optimally configure the use of the cache refreshing options in the action. For any given DirectSQLUpdate, you have the choice to refresh the cache or not. When choosing to refresh the cache you have the further option to Parse SQL. If you have multiple DirectSQLUpdate actions which update the same Oracle Transportation Management business object in a single agent, you should configure the agent to only refresh the cache on the last update. An exception to this rule is if actions in the agent itself are dependent upon the modified data.

The option to Parse SQL should be used whenever possible. By default, the refresh logic needs to refresh all child data for the business object that is being updated. For instance, a DirectSQLUpdate action in a shipment agent needs to refresh the related shipment, shipment stops, ship units, etc. When SQL parsing is enabled, the agent will attempt to determine the specific business object being updated, and only refresh this object.

Workflow Thread Tuning

Oracle Transportation Management workflow is based on a set of business topics and workflow thread groups. A Topic is the Oracle Transportation Management terminology for a particular workflow process. For example, AutoMatchInvoice is a topic. Each Topic is associated with a thread group by a glog.property. For example, the following property associates the AutoMatchInvoice Topic with the agentPlanning Thread Group.

```
glog.workflow.topic.group.glog.server.agent.business.invoice.AutoMatchI  
nvoice=agentPlanning
```

A Thread Group is a set of workflow threads dedicated to processing a set of workflow topics. Each thread group is given a number of threads that can be used to simultaneously process topics. When all threads in a group are busy processing a topic, all additional topics are placed in a queue until a thread becomes available. Oracle Transportation Management computes statistics for the amount of time a topic spends waiting to be processed and the amount of time spent processing the topic. These statistics can be reviewed with the Event Diagnostics page available on the DBA Diagnostics menu.

A long average wait time and/or a backlog of queued events may indicate a need to increase the number of threads in a thread group. It is possible to temporarily add a new thread in the Event Diagnostic page. Permanent changes to thread groups must be made with a change to the corresponding glog.property. It is important to note that properties like the number of threads for a thread group need to be explicitly removed in order to override the value. The whole key and value must match the whole key and value that is being overridden. For example:

```
!remove glog.workflow.topicGroup=agentPlanning,3  
glog.workflow.topicGroup=agentPlanning,4
```

This will change the number of agentPlanning threads from 3 to 4, assuming the default value was 3.

Adding more threads will not always increase performance. Another possibility is that some slower topics are in the same thread group as a higher volume, but short duration topic. The faster topics may be prevented from running by the slower topics. Adding more threads to this thread group may not have a significant impact on performance. However, moving the slow topics to a new thread group may improve overall system throughput by allowing the shorter topics to complete more quickly. The following properties are an example of moving a topic to a new thread group:

```
glog.workflow.topicGroup=slowEventGroup,1
```

```
glog.workflow.topic.group.glog.server.agent.business.shipment.SlowTopic
=slowEventGroup
```

When modifying workflow threads via property sets, be sure to place your properties in the `APP_WORKFLOW_THREADING` set.

Automatic Timeout for Stuck Threads

For any event queue, we can set an automatic timeout that effectively kills the thread and redirects work to a new thread of the same name. To use this feature, you need to set:

```
glog.eventQueueMonitor.polling=<polling interval in seconds>
glog.eventQueueMonitor.timeout.<queue>=<timeout in seconds>
```

The polling interval should generally be 1/2 the smallest timeout.

For example:

```
glog.eventQueueMonitor.polling=60
"glog.eventQueueMonitor.timeout.transport - SMTP"=120
"glog.eventQueueMonitor.timeout.transport - HTTP"=240
```

In this example, the system will check every minute on the health of the SMTP and HTTP threads. If an SMTP thread has been active longer than 2 minutes, it is interrupted and another SMTP thread is created in its place. If an HTTP thread has been active longer than 4 minutes, it is interrupted and another HTTP thread is created in its place.

Note: The double quotes around the property key since the queue names have spaces.

Multi-threading Bulk Plan Processing

Parts of the bulk plan process can be configured to process multiple threads simultaneously. This behavior is enabled by the following property:

```
glog.task.on=true
```

Adding more threads to these thread groups may increase the performance of a very large bulk plan. The following are the two key properties to control this functionality.

```
glog.workflow.topicGroup=planningCommit,2
glog.workflow.topicGroup=planningBuild,2
```

As described previously these values can only be overridden by using a `!remove` command in the property file before specifying the new value. Please include your changes in the "Custom Thread Properties" of the application server `glog.properties` file or the `APP_WORKFLOW_THREADING` property set. See section *Property File Management* for more information.

There are additional properties to control the behavior of the multi-threading. The default values and a description of each of these properties are available in the `glog.task.properties` file.

Database

Analyzing Tables/Gathering Statistics

The Oracle Transportation Management database performs best when optimizer statistics are gathered using scripts provided by Oracle Transportation Management. Oracle Transportation Management scripts gathers statistics by performing full compute for all the objects including indexes, columns with histograms, partitions etc.

Oracle Database also has automatic statistics gathering job GATHER_STATS_JOB, which is enabled by default when a database is created, or when a database is upgraded from an earlier database release. This job gathers statistics on all objects in the database that have missing statistics or stale statistics (stale - the underlying object has been modified significantly, i.e. more than 10% of the rows). The Scheduler runs this job during a maintenance window (by default, 10pm to 6am). This automatic job gathers statistics by estimate and not in as much detail as the Oracle Transportation Management job. Over time, it tends to override detailed statistics gathered by the Oracle Transportation Management job for fast growing tables, hence automatic job should be replaced by Oracle Transportation Management statistics job.

Below are the steps to verify that statistics are gathered by the Oracle Transportation Management statistics job.

1. In SQL*Plus, as user **GLOGOWNER**, run:

```
select min(last_analyzed) from user_tables;
```

If it returns a date older than two weeks then Oracle Transportation Management statistics job is not running since that day or it is not scheduled. In that case, schedule a weekly job using any job scheduler, and have it run <otm_install_dir>/oracle/script8/gather_table_stats.sql during a low-activity period or off-peak hours.

Note: gather_table_stats.sql also runs gather_column_histograms.sql, so both should be in same directory.

2. Disable Oracle's automatic job. In SQL*Plus, as user **SYS** (or any DBA account), run:

```
BEGIN
DBMS_AUTO_TASK_ADMIN.DISABLE (
  client_name => 'auto optimizer stats collection',
  operation => NULL,
  window_name => NULL);
END;
/
commit;
```

This will ensure Oracle Transportation Management database has up to date statistics with full compute on all objects.

Database File Storage Performance

The two most commonly used forms of database file storage are: Storage Area Network (SAN) and Network-Attached Storage (NAS). NAS provides both storage and filesystem access whereas SAN provides only block-based storage and leaves file system concerns on the client side. Typical SAN protocols are SCSI, Fibre Channel, iSCSI, ATA over Ethernet, or HyperSCSI. A NAS on the other hand communicates over Ethernet and TCP/IP. Because NAS communicates over standard network protocols there is a stronger possibility of a poorly performing network affecting the data access rate. The latency from a database server to the NAS device should be less than 150 microseconds (0.15 ms) and preferably less than 100 microseconds (0.10 ms).

Full Export and Re-import

As data is added to tables the database may need to add additional database extents to store the data and indexes. The general consensus among database administrators is that an increasing number of extents does not affect performance of the database. However, in practice performance gains have been seen by completing a Full export and import.

Checking for Stale and Partial Oracle Database Statistics

The following query can be used to check for stale schema statistics:

```
select table_name, last_analyzed,num_rows, sample_size from dba_tables
where last_analyzed is not null and owner = 'GLOGOWNER' and
last_analyzed < (SYSDATE - 7) order by last_analyzed asc
```

Statistics having a 'last_analyzed' date older than one week, should be updated.

The following query can be used to check if a partial analyze was run recently.

```
select table_name, last_analyzed,num_rows, sample_size from dba_tables
where last_analyzed is not null and owner = 'GLOGOWNER' and
sample_size < num_rows order by last_analyzed desc
```

A sample_size less than the num_rows is an indication that 'estimated' statistics were used. These tables should be re-analyzed with full statistics.

Maintaining Oracle Database Statistics

The complex queries run within Oracle Transportation Management depend heavily on the Oracle Database schema statistics. It is important that all tables and indexes are analyzed at least weekly by analyzing the whole table/index, not a sample. The Oracle 10g Database has an automatic statistics gathering job GATHER_STATS_JOB, which is enabled by default when a database is created, or when a database is upgraded from an earlier database release. This job gathers statistics on all objects in the database which have missing statistics or Stale statistics. Stale - the underlying object has been modified significantly (more than 10% of the rows). The Scheduler runs this job during a maintenance window.

This default scheduled job gathers statistics by estimate. This method does not perform the same detailed analysis as the GATHER_TABLE_STATS.sql script that is provided for Oracle Transportation Management. Over time, the Oracle 10g Database default GATHER_STATS_JOB tends to override detailed statistics gathered by Oracle Transportation Management script for fast growing tables. For this reason, the GATHER_STATS_JOB should be disabled. Please consult MetaLink note 467668.1 for more details on this subject.

Customers should schedule the Oracle Transportation Management scripts to run at least once a week. A common performance problem occurs just after a customer goes live. During this period a significant amount of data is loaded and/or created by Oracle Transportation Management. It may be necessary to monitor the growth of tables by comparing the actual number of rows in the table to the number of rows captured in the schema statistics. If the table has had significantly growth (> 10%), it is recommended that the table and any related indexes be analyzed.

To run the Oracle Transportation Management provided gather table stats script manually go to the Application servers <otm home>/glog/oracle/script8 directory. From the script8 directory log into the database as "glogowner" using SQLPlus and run the following SQL command.

```
@gather_table_stats.sql
```

Note: Clients should be warned to schedule this process to run when activity on the system is low due to potential performance impacts on the system.

SQL Tuning

Another common cause of performance issues is slow performing queries and high number of executions of queries. There are a number of mechanisms that can be used to identify these issues. The most effective is the AWR/ADDM reports provided by the Oracle Database. Another technique is to

temporarily enable SQL logging in Oracle Transportation Management and review the logs for repetitive SQL statements.

Once a poorly performing SQL statement is identified, an Explain Plan should be performed. Within Oracle Transportation Management this can be performed in SQL Servlet by pre-pending the SQL statement with "explain plan for". Performing an explain plan outside of Oracle Transportation Management will supply more information about the query. When running it outside Oracle Transportation Management, it is important to execute the stored procedure call (`vpd.set_user()`) before running the query. Otherwise the explain plan will not be evaluated with VPD applied as it would be when run within the context of Oracle Transportation Management.

Determining the Origin of a SQL Statement

SQL statements that are part of Oracle Transportation Management originate from Oracle Transportation Management base code, custom saved queries, and reports. The performance tuning of custom reports and saved queries are the responsibility of the customer. Saved queries provide a mechanism to create a query which can be used in various parts of Oracle Transportation Management including Finders (Search pages), Business Monitors, and Automation Agents. The following query can be used to search for a SQL statement which may be generated from a saved query:

```
select saved_query_gid from saved_query where sql_find_all like  
'%querytext%' or sql_check one like '%querytext %'
```

The 'querytext' should be just part of the SQL statement, since the SQL statement is slightly modified depending on where in Oracle Transportation Management the SQL statement is being used. For instance, the Business Monitor prefixes all SQL statements with "select count(*)...".

Using Timeouts with SQL Statements

To avoid performance delays associated with long-running SQL statements, Oracle Transportation Management supports optional timeouts for SQL queries, updates and stored procedure calls. These timeouts can be varied by use case.

The following properties control SQL timeouts:

```
glog.sql.query.timeout.<query use case>=<# of seconds>  
glog.sql.update.timeout.<update use case>=<# of seconds>  
glog.sql.call.timeout.<procedure use case>=<# of seconds>
```

If set to 0, the statement will never time out.

The following table summarizes valid uses cases by statement type:

Statement Type	Use Case	Description	Default, in seconds
query	uiQuery	Finders, Managers,Business monitors	300
query	agentQuery	Agent conditions, IF agent action	900
query	default	All other query statements	1800

Statement Type	Use Case	Description	Default, in seconds
update	directSqlUpdate	DIRECT SQL UPDATE agent action	1800
update	default	All update statements	1800
call	directSqlUpdate	DIRECT_SQL_UPDATE agent action	1800
call	default	All other stored procedure calls	1800

INIT.ORA Parameters

The values of INIT.ORA tuning parameters can have a significant impact on performance. The Oracle Transportation Management Installation Guide (Database Administration section) provides the latest set of recommended settings for Oracle Transportation Management. These settings are guidelines that have been established based on feedback from various customer scenarios. There is not one set of parameters that will be optimal for all installations. These parameters need to be tuned for each Oracle Transportation Management instance by an experienced Database Administrator.

Tuning a RAC Database Installation Using Services

RAC is the multi-node clustering capability for the Oracle Database. Oracle Transportation Management is certified to work in a RAC configuration. A RAC configuration will allow the database to scale, particularly in a situation where the database experiences high CPU. The default configuration of RAC is to load-balance the work across all nodes in a cluster. In some scenarios this configuration may result in significant contention between the RAC nodes due to the global buffer cache. Evidence of this issue can be seen in the AWR/ADDM reports with high wait times due to the "gc cr request" and/or "gc buffer busy" events.

An Oracle Transportation Management large, multi-threaded bulkplan process is a scenario that may experience this problem when attempting to commit a large number of shipments across multiple nodes in a cluster. It is possible to achieve a performance improvement by dedicating a particular Oracle Transportation Management process to a sub-set of the nodes in the cluster in order to reduce contention. This can be accomplished by defining services on the RAC nodes and then associating the service in the database connect string of particular nodes in the Oracle Transportation Management Application Server. This assumes that Oracle Transportation Management has been configured to use Oracle Transportation Management scalability and that particular processes are being handled by particular applications servers in the Oracle Transportation Management cluster.

The following is an example of how these services could be defined and used:

Service Name	Primary Instance(s)	Available Instance(s)
otm_oltp	rotm021, rotm022	rotm023
otm_batch	rotm023	rotm021, rotm022

To use these services, users need to configure new connect strings in the corresponding glog.properties file⁵ for each application server like the following:

otm_oltp

```
glog.database.dbaOnly.t2client.connectionURL=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=otm-rac01-vip) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP) (HOST=otm-rac02-vip) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP) (HOST=otm-rac03-vip) (PORT=1521)) (LOAD_BALANCE=yes) (CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=otm_oltp)))\
```

otm_batch

```
glog.database.dbaOnly.t2client.connectionURL=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=otm-rac01-vip) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP) (HOST=otm-rac02-vip) (PORT=1521)) (ADDRESS=(PROTOCOL=TCP) (HOST=otm-rac03-vip) (PORT=1521)) (LOAD_BALANCE=yes) (CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=otm_batch)))
```

Hash Partitioning Indexes on RAC Databases

Oracle Transportation Management installations on the RAC database may experience high cluster waits such as “gc buffer busy” on the indexes for the columns whose data is populated by database sequence generated number. Converting such indexes into hash partitioned index reduces cluster waits. Perform following steps to convert about twenty eight indexes into hash partition. Oracle Transportation Management application and web servers should be off line during this procedure.

Note: Converting regular indexes into hash partitioned index may affect range scan on the corresponding column. If user community is heavily searching column data by range then you may want to keep index same as it is.

1. Make sure all constraints are valid by running following query on the database.

```
SELECT owner, status, constraint_type, COUNT(*) FROM  
dba_constraints WHERE owner IN ('GLOGOWNER', 'REPORTOWNER') GROUP BY  
owner, status, constraint_type;
```
2. Shut down all web and application servers.
3. Go to the Application servers <otm home>/glog/oracle/script8 directory. From the script8 directory log into the database as “glogower” using SQLPlus and run the following SQL command:

```
@gen_hash_indexes.sql
```

 - This command will create script create_hash_indexes.sql in the same directory
4. Review create_hash_indexes.sql. This file will have about 28 indexes for hash partitioning. You may remove indexes which are not good candidate for hash partitioning.
5. Using SQL PLUS, log in as GLOGOWNER and run

```
@create_hash_indexes.sql
```
6. Review generated log file create_hash_indexes.log and make sure that there are no errors.

⁵ Note these properties are reserved and cannot be added via property sets.

7. Also query data dictionary to ensure that hash partitioned indexes are created, and all primary key and foreign key constraints are valid.

```
SELECT table_name, index_name FROM dba_part_indexes WHERE owner IN ('GLOGOWNER', 'REPORTOWNER');
```

```
SELECT owner, status, constraint_type, COUNT(*) FROM dba_constraints WHERE owner IN ('GLOGOWNER', 'REPORTOWNER') GROUP BY owner, status, constraint_type;
```

8. Restart all web and application servers.

Changing Database Pool Size

As your server load varies, you may need to change the number of connections in the database pool. This may increase the performance of your Oracle Transportation Management instance. To determine if you have a database connection bottleneck, follow the instructions provided under the **Microsoft Windows** section in the **Operating System Monitoring Tools** section.

After making any database pool changes, you may need to stop and restart Oracle Transportation Management, as described in the chapter **Starting and Stopping the Oracle Transportation Management Servers**. Also, before you change your pool size, consult a qualified Oracle DBA and ensure that your database can handle the new number of connections. If you set your pool connections higher than your database can support, Oracle Transportation Management will not be able to startup and function correctly. The instructions below describe how to make these changes.

1. Log into your Oracle Transportation Management system (http://<otm_webserver_name>) as the user DBA.ADMIN.
2. In the left frame expand Configuration and Administration > System Administration > Data Source Manager.
3. Select search.
4. Select the desired database pool.
5. You can now edit the initial capacity, maximum capacity, and capacity increment. If resources permit, you should set the initial and maximum capacity to the same value and save some overhead involved with expanding and shrinking the database pool.

Purging Excess Reports from the REPORT_LOG Table

Purge Process for UNIX

1. Log into the UNIX machine on which the report server is installed.
2. Run the shell script
`purgeRepLog.sh`

The script prompts for the nine parameters that are listed later in this section. Based on the parameters passed, the script deletes the records from the database and deletes the PDF files.

Note: The USERID used to logon to the UNIX machine should have delete permissions on the PDF files created by the report server.

Purge Process for Windows

1. Log into the Windows machine on which the Report server is installed.
2. Run the batch file `purgeRepLog.bat` with the parameters that are list later in this section (in the order specified).

Based on the parameters passed, the script deletes the records from the database and deletes the PDF files.

Parameters

These parameters should be passed to the procedure in the order listed below:

- Start Date – this is the starting date from which the records are to be purged. % can be used to imply all dates. Default value is NULL. The format for the field is DD-MON-YYYY.
- End Date – this date is to date till which the records are to be purged. % can be used to imply all dates. Default value is NULL. The format for the field is DD-MON-YYYY.
- Domain Name – Domain name for which the records and files are to be purged. % indicates all domains. Default value is NULL
- GL User – GL User ID for which the records and files are to be purged. '%' can be used to indicate all users. Default value is NULL
- Report GID – the Report GID for which the records and files are to be purged. % indicates all the Report GIDs. Default value is NULL
- Job Number – the Job Number for which the records and files are to be purged. % indicates all the job numbers. Default value is NULL
- DB User Name – the database user name
- DB Password – the database password
- Connect string – the database SID or the connect string. Default value is the value of the environment variable \$ORACLE_SID.

Manually Purging Transmissions from the Database

Manually purging is an alternative to truncating the partitioned tables. Learn more about the scheduled jobs that truncate partitioned tables later in the chapter. Truncating data is much faster than purging, but purging allows you to more exactly specify what data to purge.

Transmission_purge deletes data associated with xml blobs used for integrating Oracle Transportation Management with external systems. In general, the xml blobs are temporary, so there is no downside the getting rid of them.

Your DBA or System Administrator should set up a nightly or weekly background job to run purges as described below.

Manually Purge Transmissions

To purge the transmission tables:

1. Log into sql*plus as **glogowner**. (glogowner/{password}@{tnsname})
2. `sql>exec domainman.transmission_purge(<age_in days>, <total_minutes>, <batch size>);`

Note: The old procedure transmission purge "domainman.transmission_purge(<age_in days>, <where clause>,<totalminutes>,<keep status clob flag>, <batch size>);" is deprecated.

Tender collaboration records (TENDER_COLLABORATION), which have their own partition, are not able to be purged or truncated with this procedure. They can be purged with associated shipments if the parameter PURGE_TENDER_TRANSACTION=TRUE or manually purged with sql command "exec domainman.orphan_tender_purge(<total_minutes>,<batch size>);"

Transmissions with shipment events (IE_SHIPMENT_STATUS) are purged just like other elements.

The transmission purge process accepts the following arguments:

Age in days (required): the age of the transmissions that you want to purge.

If you have no idea of how many transmissions you have that are older than 30 days (for example), you can do the following query:

```
sql> select count(*) from i_transmission where sysdate-create_date >
30;
```

Total Minutes (defaults to 60): The maximum number of minutes you would like the total purge process to take. After processing a set of rows, the procedure will check the total time. If it exceeds the inputted time, the process will wrap up the existing task, and stop. This is helpful if you have a tight time schedule in which you can perform this task.

Batch size (default 5000): Limits the number of transmission records that should be processed at a time. The smaller your rollback segment, the lower this number should be set. If you get a rollback segment error, rerun the process with a lower batch size specified.

The transmission purge process deletes records from the following tables:

```
I_TRANSMISSION
I_TRANSMISSION_ACK
I_TRANSMISSION_PGROUP
I_TRANSMISSION_REFNUM
I_TRANSMISSION_REPORT
I_TRANSACTION
I_TRANSACTION_ACK
I_TRANSACTION_DETAIL
I_TRANSACTION_REFNUM
I_LOG
I_LOG_DETAIL
```

Examples:

```
exec domainman.transmission_purge(90,30,1000);
```

deletes all transmission records (for above tables) that are older than 90 days except tender. The process would run for 30 minutes and would process 1000 records at a time.

Note: Transmission_purge is an expensive operation. It should be run during off-peak hours.

Manually Purge Problem Records

If you also want to purge the PROBLEM table:

1. Log into SQL*Plus as **glogowner**. (glogowner/{password}@{tnsname})
2. sql> exec partit.purge_partit_tables(90); to purge all database records in this table older than 90 days.

Partitioned Tables

Because Oracle Transportation Management contains several integration and logging tables that can become quite large very quickly, these tables have been partitioned to allow for quick purges of older data. By partitioning the tables, a particular partition (segment) can be truncated, instead of records being individually deleted, which is inefficient for large amounts of data.

Range Partitioning

Here is an example of partitioning, or segmenting, a table where a certain year and month drive the partition the data should be placed in. This is an example of partitioning by range:

Range	Partition
Jan-Mar 2001	sales_q1
Apr-Jun 2001	sales_q2
Jul-Sep 2001	sales_q3
Oct-Dec 2001	sales_q4

```
CREATE TABLE sales
( invoice_no NUMBER,
  sale_year INT NOT NULL,
  sale_month INT NOT NULL)
PARTITION BY RANGE (sale_year, sale_month)
( PARTITION sales_q1 VALUES LESS THAN (2001, 04)
  TABLESPACE data,
  PARTITION sales_q2 VALUES LESS THAN (2001, 07)
  TABLESPACE data,
  PARTITION sales_q3 VALUES LESS THAN (2001, 10)
  TABLESPACE data,
  PARTITION sales_q4 VALUES LESS THAN (2002, 01)
  TABLESPACE data );
```

Oracle Transportation Management Partitioned Tables

The Oracle Transportation Management database has groups of partitioned tables. Most of them are range partitioned. However, the ranges are not hard-coded. Instead, a partition_key column has been added to each partitioned table, which is populated with a trigger. The value of the calculated key determines into which partition the data should be placed, based upon the INSERT_DATE. This adds to flexibility in how the tables are segmented.

For integration related partitioned tables, each table has been defined as a quarterly, a monthly, weekly, or a daily table. The frequency to which the table has been associated drives how the partitions are populated, and also the frequency of the purge. Each of these tables has four partitions (except for I_transaction and I_transmission). I_TRANSACTION has sixteen partitions, four for inbound transmissions, four for outbound transmissions, seven for mobile messages and one for tender related outbound transmissions. I_TRANSMISSION has fifteen partitions, four inbound for inbound transmissions, four for outbound transmission and seven for mobile messages.

Quarterly	Monthly	Weekly	Daily
	EXPLANATION	I_TRANSACTION (outbound)	I_TRANSACTION (mobile)
	GL_LOGIN_HISTORY	I_TRANSMISSION (outbound)	I_TRANSMISSION(mobile)
	IE_INROUTE_SERVPROV		I_MESSAGE
	IE_MARKS		I_MESSAGE_ACK

Quarterly	Monthly	Weekly	Daily
	IE_SHIPMENTSTATUS		I_LOG (mobile)
	IE_SHIPMENT_REFNUM		I_LOG_DETAIL (mobile)
	IE_SHIP_GROUP_REFNUM		I_TRANSACTION_ACK (mobile)
	IE_SS_EQUIPMENT		I_TRANSACTION_DETAIL (mobile)
	IE_SS_REMARK		I_TRANSACTION_REFNUM (mobile)
	IE_SS_STOP		I_TRANSMISSION_ACK (mobile)
	IE_SS_S_EQUIPMENT		I_TRANSMISSION_REFNUM (mobile)
	IE_S_SHIP_UNIT		I_TRANSMISSION_REPORT (mobile)
	IE_S_SHIP_UNIT_LINE		
	I_LOG		
	I_LOG_DETAIL		
	I_TRANSACTION_DETAIL		
	PROBLEM		
	I_TRANSACTION (inbound)		
	I_TRANSMISSION (inbound)		

Here is how it works:

If the table is defined as quarterly, then one quarter will go into each partition. In other words,

Quarter	Partition
Jan-Mar	partition 1
Apr-Jun	partition 2
Jul-Sep	partition 3
Oct-Dec	partition 4

If the table is monthly, then the data is segmented as follows:

Month	Partition
Jan	partition 1
Feb	partition 2
Mar	partition 3
Apr	partition 4
May	partition 1
June	partition 2
...	

If the table is weekly, then the data is divided as follows:

Week	Partition
Jan 1 -7	partition 1
Jan 8-14	partition 2
Jan 15- 21	partition 3
Jan 22-28	partition 4
Jan 29-Feb 4	partition 1
Feb 5-11	partition 2
...	

Every table reuses its partitions, because the intention is that before the end of the cycle, the oldest partition is purged in preparation for the new cycle. In other words, for a monthly table, on April 30th, partition 1 should be purged to remove January's data, which will then be used for May.

Truncate Jobs for Partitioned Tables

In order to make truncating the partitioned tables as seamless as possible, a script is provided that submits an Oracle Database job to automatically purge the oldest partition, on the last day of the cycle. These jobs should be submitted as a final step in any migration, logged in as **GLOGOWNER**.

1. Enter SET SERVEROUTPUT ON to be able to see feedback.
2. Run Exec partit.submit_job('WW');
3. Run Exec partit.submit_job('MM');
4. Run Exec partit.submit_job('Q');
5. Run Exec partit.submit_job('D');

These jobs are set to run at 1 AM on the last day of the cycle by default.

Note: I_TRANSACTION table's OUTBOUND5 partition stores tender data. OUTBOUND5 partition will not be truncated by this job; Tender data follows lifecycle of SHIPMENT data hence it will be deleted by OTM shipment purge process "mark for purge". If Tender data needs to be deleted during the shipment purge process, then change the logic configuration parameter PURGE_TENDER_TRANSACTION to TRUE from the default value of FALSE. It can also manually purged with sql command "exec domainman.orphan_tender_purge(<total_minutes>,<batch_size>);"

Schedule Jobs at a Different Time

You can specify a different time if you would like, as an optional second parameter. For example, the following command will have the job run at 2:30am. You can resubmit these jobs whenever you need to change the runtime, since it will auto-delete the previous job, and create it again with the new time.

```
SQL> exec partit.submit_job('WW','02:30');
Job 1164 has been submitted.
It will run next on 13-MAY-2002 02:30
```

Remove Jobs

If you need to remove the job, you can execute the partit.remove_job procedure:

```
SQL> EXEC PARTIT.REMOVE_JOB('WW');
PL/SQL procedure successfully completed.
```

Since the job is submitted as a standard Oracle job, you may want to read Oracle Database documentation about Oracle Database jobs, and the DBMS_JOB package that we utilize in our procedures.

If you need to manually run the purge, you can do so by running purge_partitioned_tables.sql logged in as GLOGOWNER. You will be prompted for the frequency. You may want to run this script if you are not automating the purge, or if it fails and you would like to run it manually. For example, it can look like this:

```
SQL> @purge_partitioned_tables
```

This routine will truncate the oldest partition for each table that has been identified to be purged at the inputted frequency. The valid values are 'Q' for quarterly, 'MM' for monthly, or 'WW' for weekly, or 'D' for daily. By purging the old data, it can prepare room for the next week, month, or quarter.

```
Please enter a frequency (Q, MM, WW, D): MM
```

```
Purging partition 2...
```

```
Purged partition 2 for table GL_LOGIN_HISTORY
Purged partition 2 for table IE_INROUTE_SERVPROV
Purged partition 2 for table IE_MARKS
Purged partition 2 for table IE_SHIPMENTSTATUS
Purged partition 2 for table IE_SHIPMENT_REFNUM
Purged partition 2 for table IE_SHIP_GROUP_REFNUM
Purged partition 2 for table IE_SS_EQUIPMENT
Purged partition 2 for table IE_SS_REMARK
Purged partition 2 for table IE_SS_STOP
Purged partition 2 for table IE_SS_S_EQUIPMENT
Purged partition 2 for table IE_S_SHIP_UNIT
Purged partition 2 for table IE_S_SHIP_UNIT_LINE
Purged partition 2 for table I_LOG_DETAIL
Purged partition 2 for table I_LOG
Purged partition 2 for table I_LOG_DETAIL
```

```
Purged partition 2 for table I_TRANSACTION_DETAIL
Purged partition 2 for table PROBLEM
```

```
Purged inbound i_transaction records
Purged inbound i_transmission records
```

PL/SQL procedure successfully completed.

```
SQL> @purge_partitioned_tables
```

This routine will truncate the oldest partition for each table that has been identified to be purged at the inputted frequency. The valid values are 'Q' for quarterly, 'MM' for monthly, or 'WW' for weekly, or 'D' for daily. By purging the old data, it can prepare room for the next week, month, or quarter.

Please enter a frequency (Q, MM, WW, D): D

Purging partition 3...

```
Purged partition 3 for table EXPLANATION
Purged partition I_LOG_DET_MSG_PART3 for table I_LOG_DETAIL
Purged partition I_LOG_MSG_PART3 for table I_LOG
Purged partition I_LOG_DET_MSG_PART3 for table I_LOG_DETAIL
Purged partition I_MSG_PART3 for table I_MESSAGE
Purged partition I_MSG_ACK_PART3 for table I_MESSAGE_ACK
Purged partition I_TRANSACTION_MSG_PART3 for table I_TRANSACTION
Purged partition I_TRNSCTN_ACK_MSG_PART3 for table
I_TRANSACTION_ACK
Purged partition I_TRNSCTN_DET_MSG_PART3 for table
I_TRANSACTION_DETAIL
Purged partition I_TRNSCTN_RFNM_MSG_PART3 for table
I_TRANSACTION_REFNUM
Purged partition I_TRANSMISSION_MSG_PART3 for table I_TRANSMISSION
Purged partition I_TRNSMSN_ACK_MSG_PART3 for table
I_TRANSMISSION_ACK
Purged partition I_TRNSMSN_RFNM_MSG_PART3 for table
I_TRANSMISSION_REFNUM
Purged partition I_TRNSMSN_REP_MSG_PART3 for table
I_TRANSMISSION_REPORT
```

PL/SQL procedure successfully completed.

Pinning Large Object in Memory

This section is for a production database and an experienced DBA.

Oracle Database requires space in the System Global Area (SGA) for stored packages and functions. If SGA space is fragmented, there may not be enough space to load a package or function. Oracle Database recommends pre-allocating space in the SGA shared pool for frequently used packages, procedures and functions. Pinning objects in the shared pool can improve database performance, if it is done correctly. For large packages, when pinned they do not need to be loaded and parsed from the database again and again. Pinning large package also helps to avoid ORA-04031 error (unable to allocate xxx bytes of shared memory).

It is recommended that, in an Oracle Transportation Management production database, following objects be pinned in the SGA:

```
Package:
  VPD
  PARTIT
  PKG_PURGE
  RRL
  USER_EXIT_HELPER
  RPT_GENERAL
```

Database Space Monitoring

A DBA should periodically monitor Oracle Transportation Management database(s) space usage. There are several tables containing a column with data type of CLOB. These columns can have great amount of data. Some of the CLOB tables can be loaded quite often. One example is I_TRANSMISSION and I_TRANSACTION table. These tables contain transmission/transaction's XMLs. If there are inbound /outbound transmissions in and out of system frequently, the disk space usage of these two tables can grow very quickly.

Scheduled Jobs

Oracle Transportation Management database uses Oracle Database scheduled job to perform certain activities inside the database. Some of the jobs are set up automatically after the database is installed. The others are optional and can be implemented by a DBA. An Oracle Transportation Management database DBA should monitor these jobs regularly.

Length Semantics

Oracle Transportation Management uses character semantics for new string columns and all of the PL/SQL objects to support multi-byte character encoding scheme. It is highly recommended to use character semantics during customization of Oracle Transportation Management database schemas. Change NLS_LENGTH_SEMANTICS parameter at session level before adding new column or compiling PL/SQL object during customization as shown below.

```
alter session set NLS_LENGTH_SEMANTICS=CHAR;
```

For further information about NLS_LENGTH_SEMANTICS, please refer to *Oracle Database Globalization Support Guide*.

Optimizing Application-Database Round Trips

When querying or updating a group of records, an Oracle Transportation Management application server can be configured to retrieve or send batches of records at a time. This can significantly increase perceived query performance by reducing the network round trips.

For queries, the following properties controls the SQL fetch size

```
glog.sql.query.fetch.on = true
glog.sql.query.fetch.default = 10
glog.sql.query.fetch.maximum = 250
```

The **on** property specifies whether the application should use the default JDBC fetch size. If set to true, queries will explicitly set their fetch size based on the use case. The **default** property specifies the default fetch size for application queries. Many use cases in Oracle Transportation Management, however, set a fetch size based on criteria such as page size (see the ***User Interface*** section below). In these cases, the fetch size is often controlled by implementation details and can degrade

performance. For example, that the Finder Page Size is set to 500. Aside from the delay in retrieving that many records, the overhead of having a fetch size of 500 when the number of records may be much smaller may exceed the savings in round trips. The **maximum** property provides a system-wide constraint on the fetch size.

For updates and inserts, the application can use Oracle Batch updates to minimize round trips and increase modification speed. The following properties control batch updates:

```
glog.sql.update.batch.on = true
glog.sql.update.batch.maximum = 30
```

The **on** property specifies whether the application should use the batch updates. If set to true, inserts and updates will be batched whenever possible. The **maximum** property provides a system-wide constraint on the batch size. Note that batch updating adds overhead when database operations produce errors. Since the batch updating feature cannot provide information as to which record failed the batch, on failure the application retries the update one-by-one to find the errant record.

Integration

Suppressing Persistence XML

There is a capability to suppress storing the full transmission XML in the `i_transmission` table, and the full transaction XML in the `i_transaction` table for the RemoteQuery interface. Each of these should reduce the space used by the raw XML in the database and help performance by reducing the time to persist a transmission.

The full transmission XML is suppressed by setting the following property:

```
glog.integration.transmission.suppressPersistFullXml = true
```

Note: The TransmissionHeader will be stored in the `i_transmission` table even when this property is enabled.

The Remote Query Transaction XML is suppressed by setting the following property:

```
glog.integration.transaction.suppressPersist.RemoteQuery = true
```

As of Oracle Transportation Management version 6.0, both of these properties are enabled by default.

Synchronization of Master Data

Repetitive updates of embedded elements like locations and items should be avoided since they produce unnecessary processing. Location and item masters should be synchronized independently of orders instead of embedding within the order. This practice will reduce inserts into `I_TRANSACTION` and `I_LOG` tables, thus increasing performance by reducing storage needs.

OutXMLProfile for Outbound Integration and Notification

Out XML Profiles allow you to generate outbound XML and exclude portions of outbound XML with a high degree of control. This is done by specifying what XML builder class files should be excluded when generating XML documents or by selecting an XML template on which you can base the outbound XML. Outbound XML includes integration and notification. Unnecessarily large outbound integration is a common cause of performance issues. Reducing the size of the XML will greatly reduce the load on the database and application server.

Inbound Integration Options

Large Transmission Servlet

The LargeTransmissionServlet can be used for sending exceptionally large transmission(s) into Oracle Transportation Management. The difference with WMServlet is that parsing of the XML is handled in the servlet, and there is suppression of storing the complete transmission in the database. The individual transactions are stored in the database. It persists only the individual transactions, not the transmission. Refer to the Integration Guide for more information on this subject.

Oracle Advanced Queuing

Oracle Advanced Queuing (OAQ) provides an alternate way of sending and receiving XML transmissions to/from Oracle Transportation Management. The main benefit to using OAQ is the added level of guaranteed message delivery provided by a persistent message queue. An additional benefit is the elimination of the web and application server from the processing of persisting a transmission. This has the benefit of reducing the load on the web and application servers. Refer to the Data Management Guide for details on the setup for use of Oracle Advanced Queuing.

Web Services

You can send data to Oracle Transportation Management via a web service call to the Oracle Transportation Management application server. The performance advantage of this approach is the elimination of the web tier from the process of persisting the XML. This should increase UI performance by reducing the overall load on the web server. Refer to the Integration Guide for more information on this subject.

User Interface

Finder Page Size

A Finder Page is the Oracle Transportation Management terminology for the standard result page which is returned when running an Oracle Transportation Management finder query. The number of records returned per page is configurable via the "Finder Page Size" user preference. Increasing this from the default value of 25 has a direct impact on the performance of loading this page due to the increased amount of data that needs to be processed and displayed. Values greater than 100 are likely to make the performance unacceptable.

User Favorites

Favorites allow you to limit initial Find () results so that your favorite results are displayed first. For example, you can create a favorite for locations. In that favorite you specify the top 10 locations which you use on a regular basis. Then whenever you use the Find () button the system first displays those 10 locations. If you want to find a different location you are given the option to search for more locations.

Adding Search Fields to Finders (Grid-Flattening)

Grid flattening enables administrators to present "pseudo fields" for search, result, and other manager pages that are customized to the needs of their users. Pseudo fields "flatten" a grid by presenting field labels that are more specific than the default fields. For example, a PO Number field could be added to the order base search page, so users can search for purchase orders directly by their numbers. Grid flattening is used to make it appear as if data from a child database table is on a parent database table. Using grid flattening on search and result pages should be done with caution since the resulting query is more complex and therefore susceptible to slower performance.

Active Table

Oracle Transportation Management implements data security via the Oracle Database feature Virtual Private Database (VPD). VPD provides an ability to dynamically append where conditions to SQL statements at runtime. Oracle Transportation Management uses this capability for its external predicate functionality and for cross-domain visibility. By default, a “where” condition is appended to a SQL statement for each table in the SQL. These additional where conditions can have an adverse affect on performance.

Depending upon a customer’s particular setup in Oracle Transportation Management, these where conditions may be unnecessary. A capability has been implemented to control which tables have where conditions appended when running an Oracle Transportation Management finder query. The behavior for finders is controlled by the following property:

```
glog.query.vpdActiveTables.default=value
```

The possible values are:

- none = Do not apply VPD active table.
- primary = Apply VPD active table to the primary table of the query.
- comma-delimited tables = Apply VPD active table(s) for the supplied list.

Optionally, this behavior can be overridden for a specific Oracle Transportation Management query class with a property like:

```
glog.query.vpdActiveTables.QueryClassName=value
```

For example:

```
glog.query.vpdActiveTables.glog.server.query.invoice.InvoiceQuery=primary
```

The default behavior is to use primary table as active table for all finder set queries except following queries:

```
glog.server.query.preference.QueryTableQuery  
glog.server.query.notify.StylesheetProfileQuery
```

Prior to version 6.0 the active table behavior was controlled by a property like:

```
glog.query.vpdOptimization.glog.server.query.invoice.InvoiceQuery=active
```

Status Queries

Queries based on status can be complex and often suffer from poor performance. Some of this complexity is due to the implementation of domains such that status types are created within each domain. Based on a customer’s particular domain setup it may be possible to optimize these status queries.

The following glog.property can be used to control the domains from which status types will be queried:⁶

```
glog.query.statusOptimization=value
```

⁶ Note that changes to any `glog.query` property should be set on both the web servers and application servers. If using a property set, make your changes to the common `CUSTOM` property set.

The valid values are:

- none – No optimization so all domains are used.
- currentDomain – Status types from the current domain only are used.
- childDomains – Status types from the current and child domains are used.
- grantedDomains – Status types from all granted domains are used.

The following property controls how many domains will be included before the logic reverts to the same behavior as "none":

```
glog.query.statusOptimizationDomainMax=value
```

Manager Layout Producer Configuration

The XML document for an object is created by a series of "producers". These producers are associated with objects within Oracle Transportation Management. Each manager in Oracle Transportation Management has an XML document associated with it. When you create a custom manager layout you can create or remove fields from that manager. The system creates an XML document containing relevant data that is displayed in that manager. However the XML document may contain data that is no longer needed in a custom manager that has had a lot of fields removed. By deleting the associated producers from the custom manager the XML document will be smaller, allowing the page to load more quickly. Refer to the "Manager Layout: Producers" topic in online help for more information.

Parameterized Saved Queries

The UI finder mechanism in Oracle Transportation Management is a generic engine which enables flexible querying of any business object in Oracle Transportation Management. A SQL query is generated dynamically based on the user's input in the finder query page. Due to the potential complexity of the queries and the generic nature of the engine, it is not possible to always generate SQL which will perform optimally when executed. Saved queries provided a mechanism for an end-user to create a query using custom SQL. Users can create a saved query using SQL that is formatted in a manner which performs more efficiently. This also provides a mechanism to specify a hint in the SQL which will get passed to the Oracle SQL optimizer.

In addition to a standard saved query, there is an ability to create a saved query with parameters that can be fulfilled at runtime. For more information on this topic refer to "How To Enable the Use of Parameterized Saved Queries" in the online help.

Limiting Child Records in a Manager

The Oracle Transportation Management UI architecture is based on an XML document that is created by querying a record and all of its related child records. In some scenarios the number of child records can cause the XML document to be extremely large and therefore cause high memory usage and slow performance. The number of child records returned is configurable via the glog.property:

```
glog.query.limit.glog.server.query.orderbase.QueryName=limit
```

The QueryName is the name of the corresponding query and the limit indicates the maximum number of child records to return. If the number of child records exceeds this value, the manager will not display any child records of this type.

For example:

```
glog.query.limit.glog.server.query.order.ShipUnitLineQuery=100
```

As of version 6.0 many, managers have these properties set by default.

Functional Security CheckAccess Cache Sizes

Oracle Transportation Management functional security is cached by user. If your installation is going to have a large number of simultaneous users it may be necessary to increase the size of the functional security CheckAccess cache to be equal to the peak number of users expected in order to obtain optimal performance. In the `glog.properties` file on the Oracle Transportation Management web server, or the `WEB_CUSTOM` property set, you should configure the following property. The default value is 100.

For example:

```
glog.cache.CheckAccess.capacity=nnn
```

The following property controls the number of items caches per user. It can also be adjusted as necessary. For example:

```
glog.cache.CheckAccessItem.capacity=200
```

Client Session Management

You can open multiple pages simultaneously. By default, these client sessions are managed in a "least recently used" queue, but this can be configured to have a timeout. See the About Client Session Management topic in Online Help for more information on the available configuration options.

Scalability

Oracle Transportation Management scalability can be used to solve some performance problems. However, scalability should only be used after the single application server has been tuned properly. Adding more application servers to an installation that is performing poorly may actually make the performance worse. For instance, if the network or database is the performance bottle-neck, adding more application servers will only exaggerate the issue.

One reason to add another scalability node would be that a single application server is running out of memory or performing excessive garbage collection. Since the memory that can be allocated to a single JVM is limited, adding a scalability node will expand the total available memory. In this scenario, the additional scalability node may even be on the same physical hardware (assuming it has sufficient memory and processing power). If the processing power (CPU) is being maxed out on a single application server, adding another scalability node on another physical server will increase the total processing power.

Miscellaneous

Optimizing Report Generation

Distribution of Report Components

The generation of reports in Oracle Transportation Management is delegated to the Oracle BI Publisher components. These components include:

- **Template Retrieval.** BI Publisher XML templates defining data generation, report transformation and report translation are stored in the database. The first step in report generation is their retrieval.
- **Data Generation.** Given a data generation template, BI Publisher applies input parameters to the template and constructs a data XML stream from database queries.

- **Report Transformation.** BI Publisher transforms the data XML stream and the format template into a desired output document. This document may be HTML, XML, PDF, RTF, or an Excel spreadsheet.
- **Distribution.** For ad-hoc reports, the user may request the report be served directly back to their browser. Alternatively, they may want the report to an e-mail recipient via SMTP. Scheduled reports are either sent directly via e-mail or transferred to an Oracle Transportation Management Contact. This leverages the Notification subsystem to notify interested parties of the report via e-mail, fax or message center⁷, attaching or linking to the report contents.

These components typically use significant CPU and memory resources. By default, they are run on an application server to:

- perform all data queries behind a firewall;
- avoid degradation of web server response;
- manage SMTP backup through the Oracle Transportation Management event queues;
- avoid disk storage of reports on an open web server; and
- scale report generation via application-tier scalability⁸.

There may be cases, though, where one or more of these components should run in the web-tier. For example, very large reports are more efficiently generated on the requesting web server. RMI, the transport protocol between the web servers and application servers, defaults to a maximum 10 MB message size. Trying to generate such reports on an application server may require larger RMI buffers with a subsequent impact on application-tier and web-tier memory to support the data transfer.

The following table lists properties which control report architecture:

Component Property	Description	Requirements	Default
glog.bipreports.appTier.distribute	if true, ad-hoc report e-mails are distributed via the application server; otherwise they are distributed via the web server	if false, requires SMTP server access on the web server	true
glog.bipreports.appTier.query	if true, ad-hoc report queries for templates and data are performed on the application server; otherwise, they are performed on the web server	if false, requires database access from the web server	true
glog.bipreports.appTier.transform	if true, report transformation is performed on the application server; otherwise, it is performed on the web server		true

⁷ an internal Oracle Transportation Management message system

⁸ via the new `REPORTS` application function

Note: These properties have no impact on scheduled reports: they always run on the application server.

When reports are generated on the web server, web scalability can be used to scale generation limit the impact of report generation on a core cluster of web servers. See the Application Scalability Guide for more information.

Management of Report Files

During report generation, a number of files may be stored in the server's file system. These include data content XML, embedded images for HTML and the final reports. Oracle Transportation Management must have full create/read/write/delete privileges to these files and their directories to properly stage reports.

The following properties control the location of reporting files on each tier:

Property	Holds	Default
glog.bipreports.report.path	final reports	temp
glog.bipreports.app.tempDir	data content files for the application server	temp/bipublisher
glog.bipreports.app.ImageDir	HTML embedded images for the application server	temp/bipublisher
glog.bipreports.web.tempDir	data content files for the web server	bipublisher
glog.bipreports.web.ImageDir	HTML embedded images for the web server	bipublisher9

Intermediate report files are typically removed once the final report is generated. Final report files are managed with a Least Recently Used (LRU) algorithm. The default number of reports maintained by a running server is 200. Use the following property to modify this value:

```
glog.cache.Reports.capacity=<report capacity>
```

If a server generating reports fails or is cycled, existing report files will no longer be managed by the LRU algorithm. The following set of properties provides a backup to clean out old report and image files.

Property	Holds	Default
glog.bipreports.cleanup.<type>.threshold	# of days to hold files of this type	Reports: 30 Images: 30

⁹ This must be underneath the document root to properly server images to the browser.

Property	Holds	Default
glog.bipreports.cleanup.<type>.frequency	# of minutes between file scans	Reports: 360 Images: 360
glog.bipreports.cleanup.<type>.files	comma-delimited list of file wildcards to remove	Reports: report-*, *.html, *.rtf, *.pdf, *.xls Images: xdo*

Valid types include: `content`, `report`, `webImages` and `appImages`.

Logging BI Publisher Activity

To diagnose errors during data content generation or report transformation, use the following properties to enable and control BI Publisher logging:

Property	Description	Default
glog.bipreports.log.filename	base name of BI Publisher log	bipublisher.log
glog.bipreports.log.level	detail level for BI Publisher logging. Valid values are: 6 = UNEXPECTED 5 = ERROR 4 = EXCEPTION 3 = EVENT 2 = PROCEDURE 1 = STATEMENT 0 = OFF	4
glog.bipreports.log.maxSize	maximum number of bytes in any one log file	1000000
glog.bipreports.log.numBackups	number of log backups before cycling	5

14. Troubleshooting Known Issues

Installation

There are known problems with Windows X server Terminal Emulators and InstallAnywhere on Solaris. InstallAnywhere is the installation program used by both WebLogic and Oracle Transportation Management.

Symptoms: Installer started from Windows via remote X session appears to fail immediately - or in debug mode, displays an error containing the string "An unexpected exception has been detected in native code outside the VM" and will leave a core dump in your current directory.

Resolution: Install from the server console (recommended) or log out of your X session, reset your X server and log back in.

503 (Service Unavailable) Errors

While the application server is starting up, the web server will respond to all requests with a "503 (Service Unavailable)" error message. This will go away once the application server has fully started up. See **Starting Oracle Transportation Management Servers** for more information.

Uninstalling Oracle Transportation Management

Oracle Transportation Management must be manually uninstalled; there is no automated uninstall option. Follow the directions below to completely remove Oracle Transportation Management from your system.

1. If running on Microsoft Windows, remove the Oracle Transportation Management services. The default names for these services are "Oracle Process Manager (instance1)", otmtomcat, and otmweblogic. The srvinstw.exe utility from the Microsoft Windows Resource Kit can help you remove these services.
2. Remove the Oracle Transportation Management installation directory and all of its contents (e.g. d:\gc3 on Windows and /opt/gc3 on UNIX).
3. Modify your PATH and CLASSPATH environmental variables and change them back to their original state.

Browsers

Symptoms: Your browser displays stale or incorrect data while using Oracle Transportation Management.

Resolution: Configure your browser to "Check for newer versions of stored pages: Every visit to the page". This will ensure that your browser displays the correct data and not cached pages. Depending on your browser, this option is usually under the "Cache" or "Temporary Internet files" section of the "Options" or "Preferences" window.

